1. Introduction
    1. Introduction to compressive sensing
2. Sparse and Compressible Signal Models
    1. Introduction to vector spaces
    2. Bases and frames
    3. Sparse representations
    4. Compressible signals
3. Sensing Matrices
    1. Sensing matrix design
    2. Null space conditions
    3. The restricted isometry property
    4. The RIP and the NSP
    5. Matrices that satisfy the RIP
    6. Coherence
4. Sparse Signal Recovery via ℓ_1 Minimization
    1. Signal recovery via ℓ_1 minimization
    2. Noise-free signal recovery
    3. Signal recovery in noise
    4. Instance-optimal guarantees revisited
    5. The cross-polytope and phase transitions
5. Algorithms for Sparse Recovery
    1. Sparse recovery algorithms
    2. Convex optimization-based methods
    3. Greedy algorithms
    4. Combinatorial algorithms
    5. Bayesian methods
6. Applications of Compressive Sensing
    1. Linear regression and model selection
    2. Sparse error correction
    3. Group testing and data stream algorithms
    4. Compressive medical imaging

Introduction to compressive sensing

Introduction to compressive sensing. This course introduces the basic concepts in compressive sensing. We overview the concepts of sparsity, compressibility, and transform coding. We then review applications of sparsity in several signal processing problems such as sparse recovery, model selection, data coding, and error correction. We overview the key results in these fields, focusing primarily on both theory and algorithms for sparse recovery. We also discuss applications of compressive sensing in communications, biosensing, medical imaging, and sensor networks.

We are in the midst of a digital revolution that is driving the development and deployment of new kinds of sensing systems with ever-increasing fidelity and resolution. The theoretical foundation of this revolution is the pioneering work of Kotelnikov, Nyquist, Shannon, and Whittaker on sampling continuous-time band-limited signals [link], [link], [link], [link]. Their results demonstrate that signals, images, videos, and other data can be exactly recovered from a set of uniformly spaced samples taken at the so-called *Nyquist rate* of twice the highest frequency present in the signal of interest. Capitalizing on this discovery, much of signal processing has moved from the analog to the digital domain and ridden the wave of Moore's law. Digitization has enabled the creation of sensing and processing systems that are more robust, flexible, cheaper and, consequently, more widely-used than their analog counterparts.

As a result of this success, the amount of data generated by sensing systems has grown from a trickle to a torrent. Unfortunately, in many important and emerging applications, the resulting Nyquist rate is so high that we end up with far too many samples. Alternatively, it may simply be too costly, or even physically impossible, to build devices capable of acquiring samples at the necessary rate. Thus, despite extraordinary advances in computational power, the acquisition and processing of signals in application areas such as imaging, video, medical imaging, remote surveillance, spectroscopy, and genomic data analysis continues to pose a tremendous challenge.

To address the logistical and computational challenges involved in dealing with such high-dimensional data, we often depend on compression, which aims at finding the most concise representation of a signal that is able to

achieve a target level of acceptable distortion. One of the most popular techniques for signal compression is known as *transform coding*, and typically relies on finding a basis or frame that provides *sparse* or *compressible* representations for signals in a class of interest. By a sparse representation, we mean that for a signal of length N, we can represent it with $K \ll N$ nonzero coefficients; by a compressible representation, we mean that the signal is well-approximated by a signal with only K nonzero coefficients. Both sparse and compressible signals can be represented with high fidelity by preserving only the values and locations of the largest coefficients of the signal. This process is called *sparse approximation*, and forms the foundation of transform coding schemes that exploit signal sparsity and compressibility, including the JPEG, JPEG2000, MPEG, and MP3 standards.

Leveraging the concept of transform coding, *compressive sensing* (CS) has emerged as a new framework for signal acquisition and sensor design. CS enables a potentially large reduction in the sampling and computation costs for sensing signals that have a sparse or compressible representation. The Nyquist-Shannon sampling theorem states that a certain minimum number of samples is required in order to perfectly capture an arbitrary bandlimited signal, but when the signal is sparse in a known basis we can vastly reduce the number of measurements that need to be stored. Consequently, when sensing sparse signals we might be able to do better than suggested by classical results. This is the fundamental idea behind CS: rather than first sampling at a high rate and then compressing the sampled data, we would like to find ways to *directly* sense the data in a compressed form — i.e., at a lower sampling rate. The field of CS grew out of the work of Emmanuel Candès, Justin Romberg, and Terence Tao and of David Donoho, who showed that a finite-dimensional signal having a sparse or compressible representation can be recovered from a small set of linear, nonadaptive measurements [link], [link], [link]. The design of these measurement schemes and their extensions to practical data models and acquisition schemes are one of the most central challenges in the field of CS.

Although this idea has only recently gained significant attraction in the signal processing community, there have been hints in this direction dating back as far as the eighteenth century. In 1795, Prony proposed an algorithm

for the estimation of the parameters associated with a small number of complex exponentials sampled in the presence of noise [link]. The next theoretical leap came in the early 1900's, when Carathéodory showed that a positive linear combination of *any* K sinusoids is uniquely determined by its value at $t = 0$ and at *any* other $2K$ points in time [link], [link]. This represents far fewer samples than the number of Nyquist-rate samples when K is small and the range of possible frequencies is large. In the 1990's, this work was generalized by George, Gorodnitsky, and Rao, who studied sparsity in the context of biomagnetic imaging and other contexts [link], [link], and by Bressler and Feng, who proposed a sampling scheme for acquiring certain classes of signals consisting of K components with nonzero bandwidth (as opposed to pure sinusoids) [link], [link]. In the early 2000's Vetterli, Marziliano, and Blu proposed a sampling scheme for non-bandlimited signals that are governed by only K parameters, showing that these signals can be sampled and recovered from just $2K$ samples [link].

A related problem focuses on recovery of a signal from partial observation of its Fourier transform. Beurling proposed a method for extrapolating these observations to determine the entire Fourier transform [link]. One can show that if the signal consists of a finite number of impulses, then Beurling's approach will correctly recover the entire Fourier transform (of this non-bandlimited signal) from *any* sufficiently large piece of its Fourier transform. His approach — to find the signal with smallest ℓ_1 norm among all signals agreeing with the acquired Fourier measurements — bears a remarkable resemblance to some of the algorithms used in CS.

More recently, Candès, Romberg, Tao [link], [link], [link], [link], [link], and Donoho [link] showed that a signal having a sparse representation can be recovered *exactly* from a small set of linear, nonadaptive measurements. This result suggests that it may be possible to sense sparse signals by taking far fewer measurements, hence the name *compressive* sensing. Note, however, that CS differs from classical sampling in two important respects. First, rather than sampling the signal at specific points in time, CS systems typically acquire measurements in the form of inner products between the signal and more general test functions. We will see throughout this course that *randomness* often plays a key role in the design of these test functions. Second, the two frameworks differ in the manner in which they deal with

signal recovery, i.e., the problem of recovering the original signal from the compressive measurements. In the Nyquist-Shannon framework, signal recovery is achieved through cardinal sine (sinc) interpolation — a linear process that requires little computation and has a simple interpretation.

CS has already had notable impact on several applications. One example is medical imaging, where it has enabled speedups by a factor of seven in pediatric MRI while preserving diagnostic quality [link]. Moreover, the broad applicability of this framework has inspired research that extends the CS framework by proposing practical implementations for numerous applications, including sub-Nyquist analog-to-digital converters (ADCs), compressive imaging architectures, and compressive sensor networks.

This course introduces the basic concepts in compressive sensing. We overview the concepts of sparsity, compressibility, and transform coding. We overview the key results in the field, beginning by focusing primarily on the theory of sensing matrix design, ℓ_1-minimization, and alternative algorithms for sparse recovery. We then review applications of sparsity in several signal processing problems such as sparse regression and model selection, error correction, group testing, and compressive inference. We also discuss applications of compressive sensing in analog-to-digital conversion, biosensing, conventional and hyperspectral imaging, medical imaging, and sensor networks.

Acknowledgments

The authors would like to thank Ewout van den Berg, Yonina Eldar, Piotr Indyk, Gitta Kutyniok, and Yaniv Plan for their feedback regarding some portions of this course which now also appear in the introductory chapter of *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2011.

Introduction to vector spaces
This module provides a brief review of some of the key concepts in vector spaces that will be required in developing the theory of compressive sensing.

For much of its history, signal processing has focused on signals produced by physical systems. Many natural and man-made systems can be modeled as linear. Thus, it is natural to consider signal models that complement this kind of linear structure. This notion has been incorporated into modern signal processing by modeling signals as *vectors* living in an appropriate *vector space*. This captures the linear structure that we often desire, namely that if we add two signals together then we obtain a new, physically meaningful signal. Moreover, vector spaces allow us to apply intuitions and tools from geometry in $\mathbb{R}^3$, such as lengths, distances, and angles, to describe and compare signals of interest. This is useful even when our signals live in high-dimensional or infinite-dimensional spaces.

Throughout this course, we will treat signals as real-valued functions having domains that are either continuous or discrete, and either infinite or finite. These assumptions will be made clear as necessary in each chapter. In this course, we will assume that the reader is relatively comfortable with the key concepts in vector spaces. We now provide only a brief review of some of the key concepts in vector spaces that will be required in developing the theory of compressive sensing (CS). For a more thorough review of vector spaces see this introductory course in Digital Signal Processing.

We will typically be concerned with *normed vector spaces*, i.e., vector spaces endowed with a *norm*. In the case of a discrete, finite domain, we can view our signals as vectors in an N-dimensional Euclidean space, denoted by $\mathbb{R}^N$. When dealing with vectors in $\mathbb{R}^N$, we will make frequent use of the ℓ_p norms, which are defined for $p \in [1, \infty]$ as
Equation:

$$\|x\|_p = \begin{cases} \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}}, & p \in [1, \infty); \\ \max\limits_{i=1,2,\ldots,N} |x_i|, & p = \infty. \end{cases}$$

In Euclidean space we can also consider the standard *inner product* in $\mathbb{R}^N$, which we denote

Equation:

$$\langle x, z \rangle = z^T x = \sum_{i=1}^{N} x_i z_i.$$

This inner product leads to the ℓ_2 norm: $\|x\|_2 = \sqrt{\langle x, x \rangle}$.

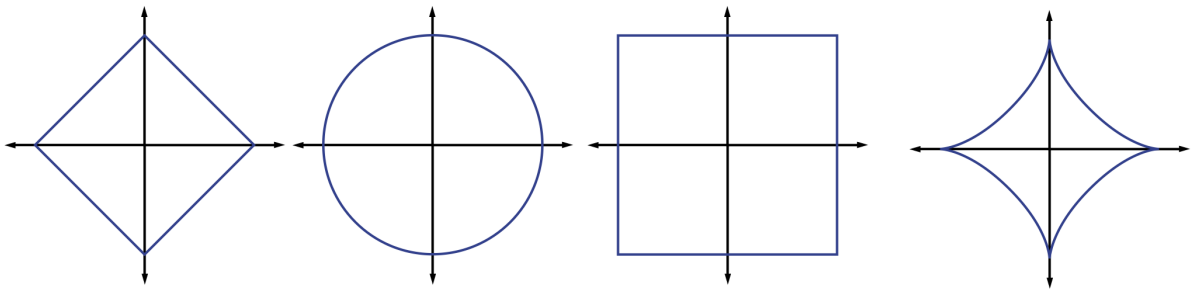
In some contexts it is useful to extend the notion of ℓ_p norms to the case where $p < 1$. In this case, the "norm" defined in [link] fails to satisfy the triangle inequality, so it is actually a quasinorm. We will also make frequent use of the notation $\|x\|_0 := |\text{supp}\,(x)|$, where $\text{supp}\,(x) = \{i : x_i \neq 0\}$ denotes the support of x and $|\text{supp}(x)|$ denotes the cardinality of $\text{supp}(x)$. Note that $\|\cdot\|_0$ is not even a quasinorm, but one can easily show that

Equation:

$$\|x\|_0 = \lim_{p \to 0} \|x\|_p^p = |\text{supp}\,(x)|,$$

justifying this choice of notation. The ℓ_p (quasi-)norms have notably different properties for different values of p. To illustrate this, in [link] we show the unit sphere, i.e., $\left\{x : \|x\|_p = 1\right\}$, induced by each of these norms in $\mathbb{R}^2$. Note that for $p < 1$ the corresponding unit sphere is nonconvex (reflecting the quasinorm's violation of the triangle inequality).

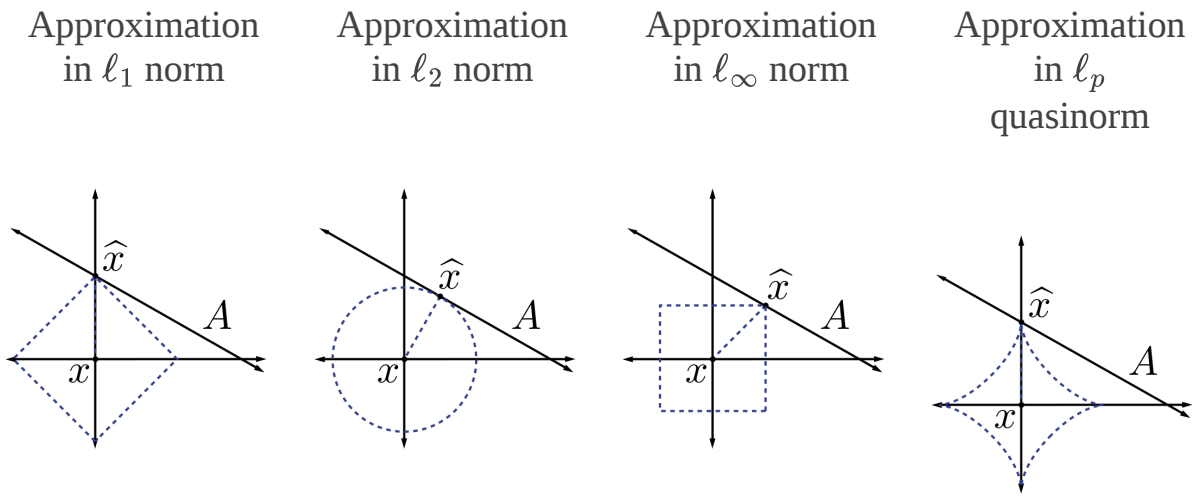
Unit sphere for ℓ_1 norm	Unit sphere for ℓ_2 norm	Unit sphere for ℓ_∞ norm	Unit sphere for ℓ_p quasinorm



Unit spheres in $\mathbb{R}^2$ for the ℓ_p norms with $p = 1, 2, \infty$, and for the ℓ_p

quasinorm with $p = \frac{1}{2}$.

We typically use norms as a measure of the strength of a signal, or the size of an error. For example, suppose we are given a signal $x \in \mathbb{R}^2$ and wish to approximate it using a point in a one-dimensional affine space A. If we measure the approximation error using an ℓ_p norm, then our task is to find the $\widehat{x} \in A$ that minimizes $\|x - \widehat{x}\|_p$. The choice of p will have a significant effect on the properties of the resulting approximation error. An example is illustrated in [link]. To compute the closest point in A to x using each ℓ_p norm, we can imagine growing an ℓ_p sphere centered on x until it intersects with A. This will be the point $\widehat{x} \in A$ that is closest to x in the corresponding ℓ_p norm. We observe that larger p tends to spread out the error more evenly among the two coefficients, while smaller p leads to an error that is more unevenly distributed and tends to be sparse. This intuition generalizes to higher dimensions, and plays an important role in the development of CS theory.



Best approximation of a point in $\mathbb{R}^2$ by a a one-dimensional subspace using the ℓ_p norms for $p = 1, 2, \infty$, and the ℓ_p quasinorm with $p = \frac{1}{2}$.

Bases and frames
This module provides an overview of bases and frames in finite-dimensional Hilbert spaces.

A set $\Psi = \{\psi_i\}_{i \in \mathscr{I}}$ is called a basis for a finite-dimensional vector space $\mathscr{V}$ if the vectors in the set span $\mathscr{V}$ and are linearly independent. This implies that each vector in the space can be represented as a linear combination of this (smaller, except in the trivial case) set of basis vectors in a unique fashion. Furthermore, the coefficients of this linear combination can be found by the inner product of the signal and a dual set of vectors. In discrete settings, we will only consider real finite-dimensional Hilbert spaces where $\mathscr{V} = \mathbb{R}^N$ and $\mathscr{I} = \{1, \ldots, N\}$.

Mathematically, any signal $x \in \mathbb{R}^N$ may be expressed as,
Equation:

$$x = \sum_{i \in \mathscr{I}} a_i \widetilde{\psi_i},$$

where our coefficients are computed as $a_i = \langle x, \psi_i \rangle$ and $\left\{\widetilde{\psi_i}\right\}_{i \in I}$ are the vectors that constitute our dual basis. Another way to denote our basis and its dual is by how they operate on x. Here, we call our dual basis $\widetilde{\Psi}$ our *synthesis basis* (used to reconstruct our signal by [link]) and Ψ is our *analysis basis*.

An orthonormal basis (ONB) is defined as a set of vectors $\Psi = \{\psi_i\}_{i \in \mathscr{I}}$ that form a basis and whose elements are orthogonal and unit norm. In other words, $\langle \psi_i, \psi_j \rangle = 0$ if $i \neq j$ and one otherwise. In the case of an ONB, the *synthesis basis* is simply the Hermitian adjoint of *analysis basis* ($\widetilde{\Psi} = \Psi^T$).

It is often useful to generalize the concept of a basis to allow for sets of possibly linearly dependent vectors, resulting in what is known as a *frame*. More formally, a frame is a set of vectors $\{\Psi_i\}_{i=1}^n$ in $\mathbb{R}^d$, $d < n$ corresponding to a matrix $\Psi \in \mathbb{R}^{d \times n}$, such that for all vectors $x \in \mathbb{R}^d$,
Equation:

$$A\|x\|_2^2 \leq \|\Psi^T x\|_2^2 \leq B\|x\|_2^2$$

with $0 < A \leq B < \infty$. Note that the condition $A > 0$ implies that the rows of Ψ must be linearly independent. When A is chosen as the largest possible value and B as the smallest for these inequalities to hold, then we call them the *(optimal) frame bounds*. If A and B can be chosen as $A = B$, then the frame is called A*-tight*, and if $A = B = 1$, then Ψ is a *Parseval frame*. A frame is called *equal-norm*, if there exists some $\lambda > 0$ such that $\|\Psi_i\|_2 = \lambda$ for all $i = 1, ..., N$, and it is *unit-norm* if $\lambda = 1$. Note also that while the concept of a frame is very general and can be defined in infinite-dimensional spaces, in the case where Ψ is a $d \times N$ matrix A and B simply correspond to the smallest and largest eigenvalues of $\Psi\Psi^T$, respectively.

Frames can provide richer representations of data due to their redundancy: for a given signal x, there exist infinitely many coefficient vectors α such that $x = \Psi\alpha$. In particular, each choice of a dual frame $\widetilde{\Psi}$ provides a different choice of a coefficient vector α. More formally, any frame satisfying
Equation:

$$\Psi\widetilde{\Psi}^T = \widetilde{\Psi}\Psi^T = I$$

is called an (alternate) dual frame. The particular choice $\widetilde{\Psi} = \left(\Psi\Psi^T\right)^{-1}\Psi$ is referred to as the *canonical dual frame*. It is also known as the Moore-Penrose pseudoinverse. Note that since $A > 0$ requires Ψ to have linearly independent rows, we ensure that $\Psi\Psi^T$ is invertible, so that $\widetilde{\Psi}$ is well-defined. Thus, one way to obtain a set of feasible coefficients is via
Equation:

$$\alpha_d = \Psi^T\left(\Psi\Psi^T\right)^{-1}x.$$

One can show that this sequence is the smallest coefficient sequence in ℓ_2 norm, i.e., $\|\alpha_d\|_2 \leq \|\alpha\|_2$ for all α such that $x = \Psi\alpha$.

Finally, note that in the sparse approximation literature, it is also common for a basis or frame to be referred to as a *dictionary* or *overcomplete dictionary* respectively, with the dictionary elements being called *atoms*.

Sparse representations

This module provides an overview of sparsity and sparse representations, giving examples for both 1-D and 2-D signals.

Transforming a signal to a new basis or frame may allow us to represent a signal more concisely. The resulting compression is useful for reducing data storage and data transmission, which can be quite expensive in some applications. Hence, one might wish to simply transmit the analysis coefficients obtained in our basis or frame expansion instead of its high-dimensional correlate. In cases where the number of non-zero coefficients is small, we say that we have a sparse representation. Sparse signal models allow us to achieve high rates of compression and in the case of compressive sensing, we may use the knowledge that our signal is sparse in a known basis or frame to recover our original signal from a small number of measurements. For sparse data, only the non-zero coefficients need to be stored or transmitted in many cases; the rest can be assumed to be zero).

Mathematically, we say that a signal x is K-*sparse* when it has at most K nonzeros, i.e., $\|x\|_0 \leq K$. We let

Equation:

$$\Sigma_K = \{x : \|x\|_0 \leq K\}$$

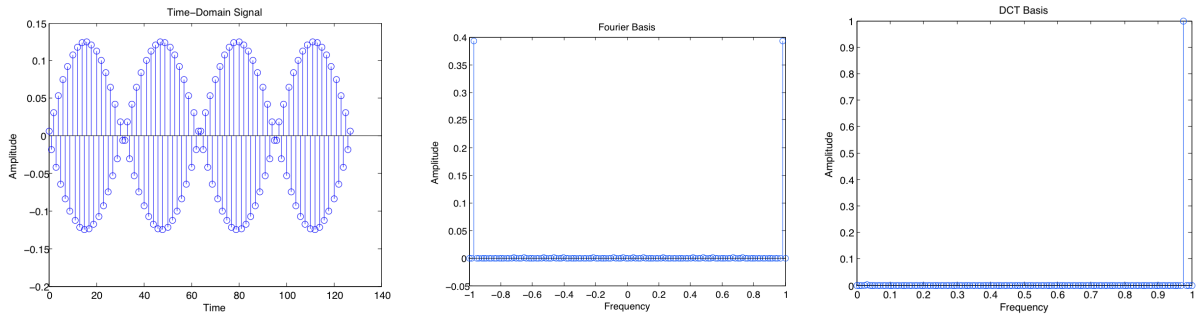
denote the set of all K-sparse signals. Typically, we will be dealing with signals that are not themselves sparse, but which admit a sparse representation in some basis Ψ. In this case we will still refer to x as being K-sparse, with the understanding that we can express x as $x = \Psi\alpha$ where $\| \alpha \|_0 \leq K$.

Sparsity has long been exploited in signal processing and approximation theory for tasks such as compression [link], [link], [link] and denoising [link], and in statistics and learning theory as a method for avoiding overfitting [link]. Sparsity also figures prominently in the theory of statistical estimation and model selection [link], [link], in the study of the human visual system [link], and has been exploited heavily in image processing tasks, since the multiscale wavelet transform [link] provides

nearly sparse representations for natural images. Below, we briefly describe some one-dimensional (1-D) and two-dimensional (2-D) examples.

1-D signal models

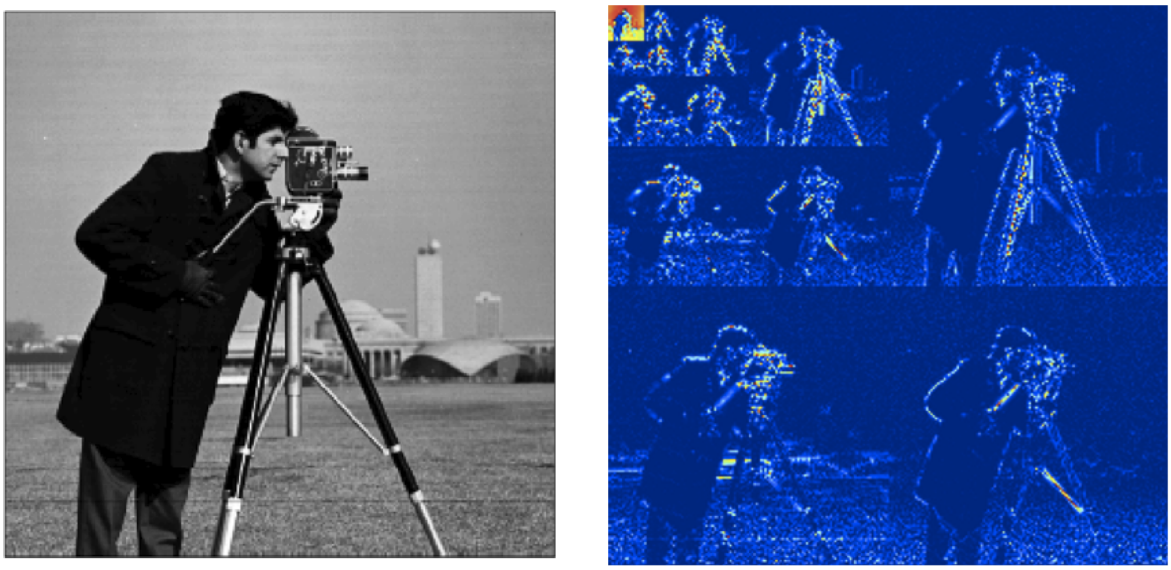
We will now present an example of three basis expansions that yield different levels of sparsity for the same signal. A simple periodic signal is sampled and represented as a periodic train of weighted impulses (see [link]). One can interpret sampling as a basis expansion where our elements in our basis are impulses placed at periodic points along the time axis. We know that in this case, our dual basis consists of sinc functions used to reconstruct our signal from discrete-time samples. This representation contains many non-zero coefficients, and due to the signal's periodicity, there are many redundant measurements. Representing the signal in the Fourier basis, on the other hand, requires only two non-zero basis vectors, scaled appropriately at the positive and negative frequencies (see [link]). Driving the number of coefficients needed even lower, we may apply the discrete cosine transform (DCT) to our signal, thereby requiring only a single non-zero coefficient in our expansion (see [link]). The DCT equation is $X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right)$ with $k = 0, \cdots, N-1$, x_n the input signal, and N the length of the transform.



Cosine signal in three representations: (a) Train of impulses (b) Fourier basis (c) DCT basis

2-D signal models

This same concept can be extended to 2-D signals as well. For instance, a binary picture of a nighttime sky is sparse in the standard pixel domain because most of the pixels are zero-valued black pixels. Likewise, natural images are characterized by large smooth or textured regions and relatively few sharp edges. Signals with this structure are known to be very nearly sparse when represented using a multiscale wavelet transform [link]. The wavelet transform consists of recursively dividing the image into its low- and high-frequency components. The lowest frequency components provide a coarse scale approximation of the image, while the higher frequency components fill in the detail and resolve edges. What we see when we compute a wavelet transform of a typical natural image, as shown in [link], is that most coefficients are very small. Hence, we can obtain a good approximation of the signal by setting the small coefficients to zero, or *thresholding* the coefficients, to obtain a K-sparse representation. When measuring the approximation error using an ℓ_p norm, this procedure yields the *best K-term approximation* of the original signal, i.e., the best approximation of the signal using only K basis elements.[footnote] Thresholding yields the best K-term approximation of a signal with respect to an orthonormal basis. When redundant frames are used, we must rely on sparse approximation algorithms like those described later in this course [link], [link].



Sparse representation of an image via a multiscale wavelet transform. (a) Original image. (b) Wavelet representation. Large coefficients are represented by light pixels, while small coefficients are represented by dark pixels. Observe that most of the wavelet coefficients are close to zero.

Sparsity results through this decomposition because in most natural images most pixel values vary little from their neighbors. Areas with little contrast difference can be represent with low frequency wavelets. Low frequency wavelets are created through stretching a mother wavelet and thus expanding it in space. On the other hand, discontinuities, or edges in the picture, require high frequency wavelets, which are created through compacting a mother wavelet. At the same time, the transitions are generally limited in space, mimicking the properties of the high frequency compacted wavelet. See "Compressible signals" for an example.

Compressible signals
This module describes compressible signals, i.e., signals that can be well-approximated by sparse signals.

Compressibility and K-term approximation

An important assumption used in the context of compressive sensing (CS) is that signals exhibit a degree of structure. So far the only structure we have considered is sparsity, i.e., the number of non-zero values the signal has when representation in an orthonormal basis Ψ. The signal is considered sparse if it has only a few nonzero values in comparison with its overall length.

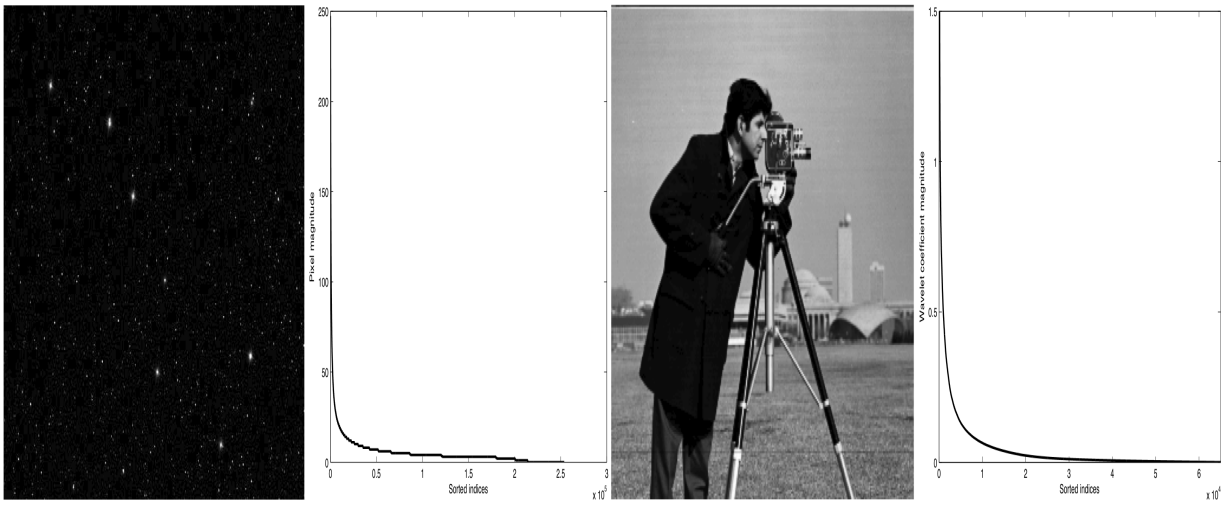
Few structured signals are truly sparse; rather they are compressible. A signal is *compressible* if its sorted coefficient magnitudes in Ψ decay rapidly. To consider this mathematically, let x be a signal which is compressible in the basis Ψ:
Equation:

$$x = \Psi\alpha,$$

where α are the coefficients of x in the basis Ψ. If x is compressible, then the magnitudes of the sorted coefficients α_s observe a power law decay:
Equation:

$$\left|\alpha_s\right| \leq C_1 s^{-q}, s = 1, 2, \ldots.$$

We define a signal as being compressible if it obeys this power law decay. The larger q is, the faster the magnitudes decay, and the more compressible a signal is. [link] shows images that are compressible in different bases.



The image in the upper left is a signal that is compressible in space. When the pixel values are sorted from largest to smallest, there is a sharp descent. The image in the lower left is not compressible in space, but it is compressible in wavelets since its wavelet coefficients exhibit a power law decay.

Because the magnitudes of their coefficients decay so rapidly, compressible signals can be represented well by $K \ll N$ coefficients. The best K-term approximation of a signal is the one in which the K largest coefficients are kept, with the rest being zero. The error between the true signal and its K term approximation is denoted the K-term approximation error $\sigma_K(x)$, defined as

Equation:

$$\sigma_K(x) = \arg\min_{\alpha \in \Sigma_K} \| x - \Psi\alpha \|_2.$$

For compressible signals, we can establish a bound with power law decay as follows:

Equation:

$$\sigma_K(x) \leq C_2 K^{1/2-s}.$$

In fact, one can show that $\sigma_K(x)_2$ will decay as K^{-r} if and only if the sorted coefficients α_i decay as $i^{-r+1/2}$ [link]. [link] shows an image and its K-term approximation.

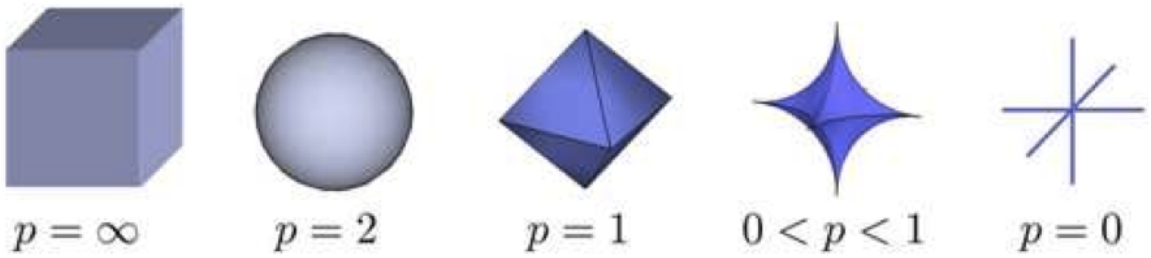
Sparse approximation of a natural image. (a) Original image. (b) Approximation of image obtained by keeping only the largest 10% of the wavelet coefficients. Because natural images are compressible in a wavelet domain, approximating this image it in terms of its largest wavelet coefficients maintains good fidelity.

Compressibility and ℓ_p spaces

A signal's compressibility is related to the ℓ_p space to which the signal belongs. An infinite sequence $x(n)$ is an element of an ℓ_p space for a particular value of p if and only if its ℓ_p norm is finite:
Equation:

$$\| x \|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}} < \infty.$$

The smaller p is, the faster the sequence's values must decay in order to converge so that the norm is bounded. In the limiting case of $p = 0$, the "norm" is actually a pseudo-norm and counts the number of non-zero values. As p decreases, the size of its corresponding ℓ_p space also decreases. [link] shows various ℓ_p unit balls (all sequences whose ℓ_p norm is 1) in 3 dimensions.



As the value of p decreases, the size of the corresponding ℓ_p space also decreases. This can be seen visually when comparing the the size of the spaces of signals, in three dimensions, for which the ℓ_p norm is less than or equal to one. The volume of these ℓ_p "balls" decreases with p.

Suppose that a signal is sampled infinitely finely, and call it $x[n]$. In order for this sequence to have a bounded ℓ_p norm, its coefficients must have a power-law rate of decay with $q > 1/p$. Therefore a signal which is in an ℓ_p space with $p \leq 1$ obeys a power law decay, and is therefore compressible.

Sensing matrix design
This module provides an overview of the sensing matrix design problem in compressive sensing.

In order to make the discussion more concrete, we will restrict our attention to the standard finite-dimensional compressive sensing (CS) model. Specifically, given a signal $x \in \mathbb{R}^N$, we consider measurement systems that acquire M linear measurements. We can represent this process mathematically as
Equation:

$$y = \Phi x,$$

where Φ is an $M \times N$ matrix and $y \in \mathbb{R}^M$. The matrix Φ represents a *dimensionality reduction*, i.e., it maps $\mathbb{R}^N$, where N is generally large, into $\mathbb{R}^M$, where M is typically much smaller than N. Note that in the standard CS framework we assume that the measurements are *non-adaptive*, meaning that the rows of Φ are fixed in advance and do not depend on the previously acquired measurements. In certain settings adaptive measurement schemes can lead to significant performance gains.

Note that although the standard CS framework assumes that x is a finite-length vector with a discrete-valued index (such as time or space), in practice we will often be interested in designing measurement systems for acquiring continuously-indexed signals such as continuous-time signals or images. For now we will simply think of x as a finite-length window of Nyquist-rate samples, and we temporarily ignore the issue of how to directly acquire compressive measurements without first sampling at the Nyquist rate.

There are two main theoretical questions in CS. First, how should we design the sensing matrix Φ to ensure that it preserves the information in the signal x? Second, how can we recover the original signal x from measurements y? In the case where our data is sparse or compressible, we will see that we can design matrices Φ with $M \ll N$ that ensure that we will be able to recover the original signal accurately and efficiently using a variety of practical algorithms.

We begin in this part of the course by first addressing the question of how to design the sensing matrix Φ. Rather than directly proposing a design procedure, we instead consider a number of desirable properties that we might wish Φ to have (including the null space property, the restricted isometry property, and bounded coherence). We then provide some important examples of matrix constructions that satisfy these properties.

Null space conditions
This module introduces the spark and the null space property, two common conditions related to the null space of a measurement matrix that ensure the success of sparse recovery algorithms. Furthermore, the null space property is shown to be a necessary condition for instance optimal or uniform recovery guarantees.

A natural place to begin in establishing conditions on Φ in the context of designing a sensing matrix is by considering the null space of Φ, denoted
Equation:

$$\mathcal{N}(\Phi) = \{z : \Phi z = 0\}.$$

If we wish to be able to recover *all* sparse signals x from the measurements Φx, then it is immediately clear that for any pair of distinct vectors $x, x' \in \Sigma_K = \{x : \|x\|_0 \leq K\}$, we must have $\Phi x \neq \Phi x'$, since otherwise it would be impossible to distinguish x from x' based solely on the measurements y. More formally, by observing that if $\Phi x = \Phi x'$ then $\Phi(x - x') = 0$ with $x - x' \in \Sigma_{2K}$, we see that Φ uniquely represents all $x \in \Sigma_K$ if and only if $\mathcal{N}(\Phi)$ contains no vectors in Σ_{2K}. There are many equivalent ways of characterizing this property; one of the most common is known as the *spark* [link].

The spark

> The spark of a given matrix Φ is the smallest number of columns of Φ that are linearly dependent.

This definition allows us to pose the following straightforward guarantee. (Corollary 1 of [link])

For any vector $y \in \mathbb{R}^M$, there exists at most one signal $x \in \Sigma_K$ such that $y = \Phi x$ if and only if $\operatorname{spark}(\Phi) > 2K$.

We first assume that, for any $y \in \mathbb{R}^M$, there exists at most one signal $x \in \Sigma_K$ such that $y = \Phi x$. Now suppose for the sake of a contradiction that $\operatorname{spark}(\Phi) \leq 2K$. This means that there exists some set of at most $2K$

columns that are linearly dependent, which in turn implies that there exists an $h \in \mathcal{N}(\Phi)$ such that $h \in \Sigma_{2K}$. In this case, since $h \in \Sigma_{2K}$ we can write $h = x - x'$, where $x, x' \in \Sigma_K$. Thus, since $h \in \mathcal{N}(\Phi)$ we have that $\Phi(x - x') = 0$ and hence $\Phi x = \Phi x'$. But this contradicts our assumption that there exists at most one signal $x \in \Sigma_K$ such that $y = \Phi x$. Therefore, we must have that $\text{spark}(\Phi) > 2K$.

Now suppose that $\text{spark}(\Phi) > 2K$. Assume that for some y there exist $x, x' \in \Sigma_K$ such that $y = \Phi x = \Phi x'$. We therefore have that $\Phi(x - x') = 0$. Letting $h = x - x'$, we can write this as $\Phi h = 0$. Since $\text{spark}(\Phi) > 2K$, all sets of up to $2K$ columns of Φ are linearly independent, and therefore $h = 0$. This in turn implies $x = x'$, proving the theorem.

It is easy to see that $\text{spark}(\Phi) \in [2, M + 1]$. Therefore, [link] yields the requirement $M \geq 2K$.

The null space property

When dealing with *exactly* sparse vectors, the spark provides a complete characterization of when sparse recovery is possible. However, when dealing with *approximately* sparse signals we must introduce somewhat more restrictive conditions on the null space of Φ [link]. Roughly speaking, we must also ensure that $\mathcal{N}(\Phi)$ does not contain any vectors that are too compressible in addition to vectors that are sparse. In order to state the formal definition we define the following notation that will prove to be useful throughout much of this course. Suppose that $\Lambda \subset \{1, 2, \cdots, N\}$ is a subset of indices and let $\Lambda^c = \{1, 2, \cdots, N\} \setminus \Lambda$. By x_Λ we typically mean the length N vector obtained by setting the entries of x indexed by Λ^c to zero. Similarly, by Φ_Λ we typically mean the $M \times N$ matrix obtained by setting the columns of Φ indexed by Λ^c to zero.[footnote]
We note that this notation will occasionally be abused to refer to the length $|\Lambda|$ vector obtained by keeping only the entries corresponding to Λ or the $M \times |\Lambda|$ matrix obtained by only keeping the columns corresponding to Λ. The usage should be clear from the context, but typically there is no substantive difference between the two.

A matrix Φ satisfies the *null space property* (NSP) of order K if there exists a constant $C > 0$ such that,
Equation:

$$\|h_\Lambda\|_2 \leq C \frac{\|h_{\Lambda^c}\|_1}{\sqrt{K}}$$

holds for all $h \in \mathcal{N}(\Phi)$ and for all Λ such that $|\Lambda| \leq K$.

The NSP quantifies the notion that vectors in the null space of Φ should not be too concentrated on a small subset of indices. For example, if a vector h is exactly K-sparse, then there exists a Λ such that $\|h_{\Lambda^c}\|_1 = 0$ and hence [link] implies that $h_\Lambda = 0$ as well. Thus, if a matrix Φ satisfies the NSP then the only K-sparse vector in $\mathcal{N}(\Phi)$ is $h = 0$.

To fully illustrate the implications of the NSP in the context of sparse recovery, we now briefly discuss how we will measure the performance of sparse recovery algorithms when dealing with general non-sparse x. Towards this end, let $\Delta : \mathbb{R}^M \to \mathbb{R}^N$ represent our specific recovery method. We will focus primarily on guarantees of the form
Equation:

$$\|\Delta(\Phi x) - x\|_2 \leq C \frac{\sigma_K(x)_1}{\sqrt{K}}$$

for all x, where we recall that
Equation:

$$\sigma_K(x)_p = \min_{\widehat{x} \in \Sigma_K} \|x - \widehat{x}\|_p.$$

This guarantees exact recovery of all possible K-sparse signals, but also ensures a degree of robustness to non-sparse signals that directly depends on how well the signals are approximated by K-sparse vectors. Such guarantees are called *instance-optimal* since they guarantee optimal performance for each instance of x [link]. This distinguishes them from guarantees that only hold

for some subset of possible signals, such as sparse or compressible signals — the quality of the guarantee adapts to the particular choice of x. These are also commonly referred to as *uniform guarantees* since they hold uniformly for all x.

Our choice of norms in [link] is somewhat arbitrary. We could easily measure the reconstruction error using other ℓ_p norms. The choice of p, however, will limit what kinds of guarantees are possible, and will also potentially lead to alternative formulations of the NSP. See, for instance, [link]. Moreover, the form of the right-hand-side of [link] might seem somewhat unusual in that we measure the approximation error as $\sigma_K(x)_1/\sqrt{K}$ rather than simply something like $\sigma_K(x)_2$. However, we will see later in this course that such a guarantee is actually not possible without taking a prohibitively large number of measurements, and that [link] represents the best possible guarantee we can hope to obtain (see "Instance-optimal guarantees revisited").

Later in this course, we will show that the NSP of order $2K$ is sufficient to establish a guarantee of the form [link] for a practical recovery algorithm (see "Noise-free signal recovery"). Moreover, the following adaptation of a theorem in [link] demonstrates that if there exists *any* recovery algorithm satisfying [link], then Φ must necessarily satisfy the NSP of order $2K$.
(Theorem 3.2 of [link])

Let $\Phi : \mathbb{R}^N \to \mathbb{R}^M$ denote a sensing matrix and $\Delta : \mathbb{R}^M \to \mathbb{R}^N$ denote an arbitrary recovery algorithm. If the pair (Φ, Δ) satisfies [link] then Φ satisfies the NSP of order $2K$.

Suppose $h \in \mathcal{N}(\Phi)$ and let Λ be the indices corresponding to the $2K$ largest entries of h. We next split Λ into Λ_0 and Λ_1, where $|\Lambda_0|=|\Lambda_1|= K$. Set $x = h_{\Lambda_1} + h_{\Lambda^c}$ and $x' = -h_{\Lambda_0}$, so that $h = x - x'$. Since by construction $x' \in \Sigma_K$, we can apply [link] to obtain $x' = \Delta\left(\Phi x'\right)$. Moreover, since $h \in \mathcal{N}(\Phi)$, we have
Equation:

$$\Phi h = \Phi\big(x - x'\big) = 0$$

so that $\Phi x' = \Phi x$. Thus, $x' = \Delta\left(\Phi x\right)$. Finally, we have that

Equation:

$$\|h_\Lambda\|_2 \leq \|h\|_2 = \|x - x'\|_2 = \|x - \Delta(\Phi x)\|_2 \leq C \frac{\sigma_K(x)_1}{\sqrt{K}} = \sqrt{2} C \frac{\|h_{\Lambda^c}\|_1}{\sqrt{2K}},$$

where the last inequality follows from [link].

The restricted isometry property
In this module we introduce the restricted isometry property (RIP) and discuss its role in compressive sensing. In particular, we describe the relationship between the RIP and the concept of stability in the context of sparse signal acquisition. We also provide a simple lower bound on the number of measurements necessary for a matrix to satisfy the RIP.

The null space property (NSP) is both necessary and sufficient for establishing guarantees of the form
Equation:

$$\|\Delta(\Phi x) - x\|_2 \leq C\frac{\sigma_K(x)_1}{\sqrt{K}},$$

but these guarantees do not account for *noise*. When the measurements are contaminated with noise or have been corrupted by some error such as quantization, it will be useful to consider somewhat stronger conditions. In [link], Candès and Tao introduced the following isometry condition on matrices Φ and established its important role in compressive sensing (CS).

> A matrix Φ satisfies the *restricted isometry property* (RIP) of order K if there exists a $\delta_K \in (0,1)$ such that
> **Equation:**
>
> $$(1-\delta_K)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\delta_K)\|x\|_2^2,$$
>
> holds for all $x \in \Sigma_K = \{x : \|x\|_0 \leq K\}$.

If a matrix Φ satisfies the RIP of order $2K$, then we can interpret [link] as saying that Φ approximately preserves the distance between any pair of K-sparse vectors. This will clearly have fundamental implications concerning robustness to noise.

It is important to note that in our definition of the RIP we assume bounds that are symmetric about 1, but this is merely for notational convenience. In practice, one could instead consider arbitrary bounds
Equation:

$$\alpha\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq \beta\|x\|_2^2$$

where $0 < \alpha \leq \beta < \infty$. Given any such bounds, one can always scale Φ so that it satisfies the symmetric bound about 1 in [link]. Specifically, multiplying Φ by $\sqrt{2/(\beta+\alpha)}$ will result in an $\widetilde{\Phi}$ that satisfies [link] with constant $\delta_K = (\beta - \alpha)/(\beta + \alpha)$. We will not explicitly show this, but one can check that all of the theorems in this course based on the assumption that Φ satisfies the RIP actually hold as long as there exists some scaling of Φ that satisfies the RIP. Thus, since we can always scale Φ to satisfy [link], we lose nothing by restricting our attention to this simpler bound.

Note also that if Φ satisfies the RIP of order K with constant δ_K, then for any $K' < K$ we automatically have that Φ satisfies the RIP of order K' with constant $\delta_{K'} \leq \delta_K$. Moreover, in [link] it is shown that if Φ satisfies the RIP of order K with a sufficiently small constant, then it will also automatically satisfy the RIP of order γK for certain γ, albeit with a somewhat worse constant.
(Corollary 3.4 of [link])

Suppose that Φ satisfies the RIP of order K with constant δ_K. Let γ be a positive integer. Then Φ satisfies the RIP of order $K' = \gamma\left\lfloor \frac{K}{2} \right\rfloor$ with constant $\delta_{K'} < \gamma \cdot \delta_K$, where $\lfloor \cdot \rfloor$ denotes the floor operator.

This lemma is trivial for $\gamma = 1, 2$, but for $\gamma \geq 3$ (and $K \geq 4$) this allows us to extend from RIP of order K to higher orders. Note however, that δ_K must be sufficiently small in order for the resulting bound to be useful.

The RIP and stability

We will see later in this course that if a matrix Φ satisfies the RIP, then this is sufficient for a variety of algorithms to be able to successfully recover a sparse signal from noisy measurements. First, however, we will take a closer look at whether the RIP is actually necessary. It should be clear that the lower bound in the RIP is a necessary condition if we wish to be able to recover all sparse signals x from the measurements Φx for the same reasons that the NSP is necessary. We can say even more about the necessity of the RIP by considering the following notion of stability.

> Let $\Phi : \mathbb{R}^N \to \mathbb{R}^M$ denote a sensing matrix and $\Delta : \mathbb{R}^M \to \mathbb{R}^N$ denote a recovery algorithm. We say that the pair (Φ, Δ) is *C-stable* if for any $x \in \Sigma_K$ and any $e \in \mathbb{R}^M$ we have that
> **Equation:**

$$\|\Delta(\Phi x + e) - x\|_2 \leq C\|e\|.$$

This definition simply says that if we add a small amount of noise to the measurements, then the impact of this on the recovered signal should not be arbitrarily large. [link] below demonstrates that the existence of any decoding algorithm (potentially impractical) that can stably recover from noisy measurements requires that Φ satisfy the lower bound of [link] with a constant determined by C.

If the pair (Φ, Δ) is C-stable, then
Equation:

$$\frac{1}{C}\|x\|_2 \leq \|\Phi x\|_2$$

for all $x \in \Sigma_{2K}$.

Pick any $x, z \in \Sigma_K$. Define
Equation:

$$e_x = \frac{\Phi(z - x)}{2} \qquad \text{and} \qquad e_z = \frac{\Phi(x - z)}{2},$$

and note that
Equation:

$$\Phi x + e_x = \Phi z + e_z = \frac{\Phi(x + z)}{2}.$$

Let $\widehat{x} = \Delta\left(\Phi x + e_x\right) = \Delta\left(\Phi z + e_z\right)$. From the triangle inequality and the definition of C-stability, we have that
Equation:

$$\begin{aligned} \|x - z\|_2 &= \|x - \widehat{x} + \widehat{x} - z\|_2 \\ &\leq \|x - \widehat{x}\|_2 + \|\widehat{x} - z\|_2 \\ &\leq C\|e_x\| + C\|e_z\|_2 \\ &= C\|\Phi x - \Phi z\|_2. \end{aligned}$$

Since this holds for any $x, z \in \Sigma_K$, the result follows.

Note that as $C \to 1$, we have that Φ must satisfy the lower bound of [link] with $\delta_K = 1 - 1/C^2 \to 0$. Thus, if we desire to reduce the impact of noise in our recovered signal then we must adjust Φ so that it satisfies the lower bound of [link] with a tighter constant.

One might respond to this result by arguing that since the upper bound is not necessary, we can avoid redesigning Φ simply by rescaling Φ so that as long as Φ satisfies the RIP with $\delta_{2K} < 1$, the rescaled version $\alpha\Phi$ will satisfy [link] for any constant C. In settings where the size of the noise is independent of our choice of Φ, this is a valid point — by scaling Φ we are simply adjusting the gain on the "signal" part of our measurements, and if increasing this gain does not impact the noise, then we can achieve arbitrarily high signal-to-noise ratios, so that eventually the noise is negligible compared to the signal.

However, in practice we will typically not be able to rescale Φ to be arbitrarily large. Moreover, in many practical settings the noise is not independent of Φ. For example, suppose that the noise vector e represents quantization noise produced by a finite dynamic range quantizer with B bits. Suppose the measurements lie in the interval $[-T, T]$, and we have adjusted the quantizer to capture this range. If we rescale Φ by α, then the measurements now lie between $[-\alpha T, \alpha T]$, and we must scale the dynamic range of our quantizer by α. In this case the resulting quantization error is simply αe, and we have achieved *no reduction* in the reconstruction error.

Measurement bounds

We can also consider how many measurements are necessary to achieve the RIP. If we ignore the impact of δ and focus only on the dimensions of the problem (N, M, and K) then we can establish a simple lower bound. We first provide a preliminary lemma that we will need in the proof of the main theorem.

Let K and N satisfying $K < N/2$ be given. There exists a set $X \subset \Sigma_K$ such that for any $x \in X$ we have $\|x\|_2 \leq \sqrt{K}$ and for any $x, z \in X$ with $x \neq z$,
Equation:

$$\|x - z\|_2 \geq \sqrt{K/2},$$

and
Equation:

$$\log |X| \geq \frac{K}{2} \log \left(\frac{N}{K} \right).$$

We will begin by considering the set
Equation:

$$U = \left\{ x \in \{0, +1, -1\}^N : \|x\|_0 = K \right\}.$$

By construction, $\|x\|_2^2 = K$ for all $x \in U$. Thus if we construct X by picking elements from U then we automatically have $\|x\|_2 \leq \sqrt{K}$.

Next, observe that $|U| = \binom{N}{K} 2^K$. Note also that $\|x - z\|_0 \leq \|x - z\|_2^2$, and thus if $\|x - z\|_2^2 \leq K/2$ then $\|x - z\|_0 \leq K/2$. From this we observe that for any fixed $x \in U$,
Equation:

$$\left| \left\{ z \in U : \|x - z\|_2^2 \leq K/2 \right\} \right| \leq |\{ z \in U : \|x - z\|_0 \leq K/2 \}| \leq \binom{N}{K/2} 3^{K/2}.$$

Thus, suppose we construct the set X by iteratively choosing points that satisfy [link]. After adding j points to the set, there are at least
Equation:

$$\binom{N}{K} 2^K - j \binom{N}{K/2} 3^{K/2}$$

points left to pick from. Thus, we can construct a set of size $|X|$ provided that
Equation:

$$|X| \binom{N}{K/2} 3^{K/2} \leq \binom{N}{K} 2^K$$

Next, observe that
Equation:

$$\frac{\binom{N}{K}}{\binom{N}{K/2}} = \frac{(K/2)!(N-K/2)!}{K!(N-K)!} = \prod_{i=1}^{K/2} \frac{N-K+i}{K/2+i} \geq \left(\frac{N}{K} - \frac{1}{2}\right)^{K/2},$$

where the inequality follows from the fact that $(n-K+i)/(K/2+i)$ is decreasing as a function of i. Thus, if we set $|X| = (N/K)^{K/2}$ then we have
Equation:

$$|X|\left(\frac{3}{4}\right)^{K/2} = \left(\frac{3N}{4K}\right)^{K/2} = \left(\frac{N}{K} - \frac{N}{4K}\right)^{K/2} \leq \left(\frac{N}{K} - \frac{1}{2}\right)^{K/2} \leq \frac{\binom{N}{K}}{\binom{N}{K/2}}.$$

Hence, [link] holds for $|X| = (N/K)^{K/2}$, which establishes the lemma.

Using this lemma, we can establish the following bound on the required number of measurements to satisfy the RIP.

Let Φ be an $M \times N$ matrix that satisfies the RIP of order $2K$ with constant $\delta \in \left(0, \frac{1}{2}\right]$. Then
Equation:

$$M \geq CK \log\left(\frac{N}{K}\right)$$

where $C = 1/2 \log\left(\sqrt{24}+1\right) \approx 0.28$.

We first note that since Φ satisfies the RIP, then for the set of points X in [link] we have,
Equation:

$$\|\Phi x - \Phi z\|_2 \geq \sqrt{1-\delta}\|x-z\|_2 \geq \sqrt{K/4}$$

for all $x, z \in X$, since $x - z \in \Sigma_{2K}$ and $\delta \leq \frac{1}{2}$. Similarly, we also have
Equation:

$$\|\Phi x\|_2 \leq \sqrt{1+\delta}\|x\|_2 \leq \sqrt{3K/2}$$

for all $x \in X$.

From the lower bound we can say that for any pair of points $x, z \in X$, if we center balls of radius $\sqrt{K/4}/2 = \sqrt{K/16}$ at Φx and Φz, then these balls will be disjoint. In turn, the upper bound tells us that the entire set of balls is itself contained within a larger ball of radius $\sqrt{3K/2} + \sqrt{K/16}$. If we let $B^M(r) = \left\{x \in \mathbb{R}^M : \|x\|_2 \leq r\right\}$, this implies that
Equation:

$$\begin{aligned}
\mathrm{Vol}\left(B^M\left(\sqrt{3K/2} + \sqrt{K/16}\right)\right) &\geq |X| \cdot \mathrm{Vol}\left(B^M\left(\sqrt{K/16}\right)\right), \\
\left(\sqrt{3K/2} + \sqrt{K/16}\right)^M &\geq |X| \cdot \left(\sqrt{K/16}\right)^M, \\
\left(\sqrt{24} + 1\right)^M &\geq |X|, \\
M &\geq \frac{\log|X|}{\log\left(\sqrt{24}+1\right)}.
\end{aligned}$$

The theorem follows by applying the bound for $|X|$ from [link].

Note that the restriction to $\delta \leq \frac{1}{2}$ is arbitrary and is made merely for convenience — minor modifications to the argument establish bounds for $\delta \leq \delta_{\max}$ for any $\delta_{\max} < 1$. Moreover, although we have made no effort to optimize the constants, it is worth noting that they are already quite reasonable.

Although the proof is somewhat less direct, one can establish a similar result (in the dependence on N and K) by examining the *Gelfand width* of the ℓ_1 ball [link]. However, both this result and [link] fail to capture the precise dependence of M on the desired RIP constant δ. In order to quantify this dependence, we can exploit recent results concerning the *Johnson-Lindenstrauss lemma*, which concerns embeddings of finite sets of points in low-dimensional spaces [link]. Specifically, it

is shown in [link] that if we are given a point cloud with p points and wish to embed these points in $\mathbb{R}^M$ such that the squared ℓ_2 distance between any pair of points is preserved up to a factor of $1 \pm \epsilon$, then we must have that
Equation:

$$M \geq \frac{c_0 \log(p)}{\epsilon^2},$$

where $c_0 > 0$ is a constant.

The Johnson-Lindenstrauss lemma is closely related to the RIP. We will see in "Matrices that satisfy the RIP" that any procedure that can be used for generating a linear, distance-preserving embedding for a point cloud can also be used to construct a matrix that satisfies the RIP. Moreover, in [link] it is shown that if a matrix Φ satisfies the RIP of order $K = c_1 \log(p)$ with constant δ, then Φ can be used to construct a distance-preserving embedding for p points with $\epsilon = \delta/4$. Combining these we obtain
Equation:

$$M \geq \frac{c_0 \log(p)}{\epsilon^2} = \frac{16 c_0 K}{c_1 \delta^2}.$$

Thus, for small δ the number of measurements required to ensure that Φ satisfies the RIP of order K will be proportional to K/δ^2, which may be significantly higher than $K \log(N/K)$. See [link] for further details.

The RIP and the NSP
This module describes the relationship between the restricted isometry property (RIP) and the null space property (NSP). Specifically, it is shown that a matrix which satisfies the RIP will also satisfy the NSP.

Next we will show that if a matrix satisfies the restricted isometry property (RIP), then it also satisfies the null space property (NSP). Thus, the RIP is strictly stronger than the NSP.

Suppose that Φ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$. Then Φ satisfies the NSP of order $2K$ with constant
Equation:

$$C = \frac{\sqrt{2}\delta_{2K}}{1 - \left(1 + \sqrt{2}\right)\delta_{2K}}.$$

The proof of this theorem involves two useful lemmas. The first of these follows directly from standard norm inequality by relating a K-sparse vector to a vector in $\mathbb{R}^K$. We include a simple proof for the sake of completeness.

Suppose $u \in \Sigma_K$. Then
Equation:

$$\frac{\|u\|_1}{\sqrt{K}} \leq \|u\|_2 \leq \sqrt{K}\|u\|_\infty.$$

For any u, $\|u\|_1 = |\langle u, \operatorname{sgn}(u)\rangle|$. By applying the Cauchy-Schwarz inequality we obtain $\|u\|_1 \leq \|u\|_2 \|\operatorname{sgn}(u)\|_2$. The lower bound follows since $\operatorname{sgn}(u)$ has exactly K nonzero entries all equal to ± 1 (since $u \in \Sigma_K$) and thus $\|\operatorname{sgn}(u)\| = \sqrt{K}$. The upper bound is obtained by observing that each of the K nonzero entries of u can be upper bounded by $\|u\|_\infty$.

Below we state the second key lemma that we will need in order to prove [link]. This result is a general result which holds for arbitrary h, not just vectors $h \in \mathcal{N}(\Phi)$. It should be clear that when we do have $h \in \mathcal{N}(\Phi)$, the argument could be simplified considerably. However, this lemma will prove immensely useful when we turn to the problem of sparse recovery from noisy measurements later in this course, and thus we establish it now in its full generality. We state the lemma here, which is proven in "ℓ_1 minimization proof".

Suppose that Φ satisfies the RIP of order $2K$, and let $h \in \mathbb{R}^N$, $h \neq 0$ be arbitrary. Let Λ_0 be any subset of $\{1, 2, ..., N\}$ such that $|\Lambda_0| \leq K$. Define Λ_1 as the index set corresponding to the K entries of $h_{\Lambda_0^c}$ with largest magnitude, and set $\Lambda = \Lambda_0 \cup \Lambda_1$. Then
Equation:

$$\|h_\Lambda\|_2 \leq \alpha \frac{\|h_{\Lambda_0^c}\|_1}{\sqrt{K}} + \beta \frac{|\langle \Phi h_\Lambda, \Phi h \rangle|}{\|h_\Lambda\|_2},$$

where
Equation:

$$\alpha = \frac{\sqrt{2}\delta_{2K}}{1 - \delta_{2K}}, \quad \beta = \frac{1}{1 - \delta_{2K}}.$$

Again, note that [link] holds for arbitrary h. In order to prove [link], we merely need to apply [link] to the case where $h \in \mathcal{N}(\Phi)$.

Towards this end, suppose that $h \in \mathcal{N}(\Phi)$. It is sufficient to show that
Equation:

$$\|h_\Lambda\|_2 \leq C \frac{\|h_{\Lambda^c}\|_1}{\sqrt{K}}$$

holds for the case where Λ is the index set corresponding to the $2K$ largest entries of h. Thus, we can take Λ_0 to be the index set corresponding to the K largest entries of h and apply [link].

The second term in [link] vanishes since $\Phi h = 0$, and thus we have
Equation:

$$\|h_\Lambda\|_2 \leq \alpha \frac{\|h_{\Lambda_0^c}\|_1}{\sqrt{K}}.$$

Using [link],
Equation:

$$\|h_{\Lambda_0^c}\|_1 = \|h_{\Lambda_1}\|_1 + \|h_{\Lambda^c}\|_1 \leq \sqrt{K}\|h_{\Lambda_1}\|_2 + \|h_{\Lambda^c}\|_1$$

resulting in
Equation:

$$\|h_\Lambda\|_2 \leq \alpha\left(\|h_{\Lambda_1}\|_2 + \frac{\|h_{\Lambda^c}\|_1}{\sqrt{K}}\right).$$

Since $\|h_{\Lambda_1}\|_2 \leq \|h_\Lambda\|_2$, we have that
Equation:

$$(1-\alpha)\|h_\Lambda\|_2 \leq \alpha \frac{\|h_{\Lambda^c}\|_1}{\sqrt{K}}.$$

The assumption $\delta_{2K} < \sqrt{2} - 1$ ensures that $\alpha < 1$, and thus we may divide by $1 - \alpha$ without changing the direction of the inequality to establish [link] with constant
Equation:

$$C = \frac{\alpha}{1-\alpha} = \frac{\sqrt{2}\delta_{2K}}{1-\left(1+\sqrt{2}\right)\delta_{2K}},$$

as desired.

Matrices that satisfy the RIP
This module provides some examples of matrices that satisfy the restricted isometry property (RIP), focusing primarily on random constructions.

We now turn to the question of how to construct matrices that satisfy the restricted isometry property (RIP). It is possible to deterministically construct matrices of size $M \times N$ that satisfy the RIP of order K, but such constructions also require M to be relatively large [link], [link]. For example, the construction in [link] requires $M = O\left(K^2 \log N\right)$ while the construction in [link] requires $M = O(KN^{\alpha})$ for some constant α. In many real-world settings, these results would lead to an unacceptably large requirement on M.

Fortunately, these limitations can be overcome by randomizing the matrix construction. We will construct our random matrices as follows: given M and N, generate random matrices Φ by choosing the entries φ_{ij} as independent realizations from some probability distribution. We begin by observing that if all we require is that $\delta_{2K} > 0$ then we may set $M = 2K$ and draw a Φ according to a Gaussian distribution. With probability 1, any subset of $2K$ columns will be linearly independent, and hence all subsets of $2K$ columns will be bounded below by $1 - \delta_{2K}$ where $\delta_{2K} > 0$. However, suppose we wish to know the constant δ_{2K}. In order to find the value of the constant we must consider all possible $\binom{N}{K}$ K-dimensional subspaces of $\mathbb{R}^N$. From a computational perspective, this is impossible for any realistic values of N and K. Thus, we focus on the problem of achieving the RIP of order $2K$ for a specified constant δ_{2K}. Our treatment is based on the simple approach first described in [link] and later generalized to a larger class of random matrices in [link].

To ensure that the matrix will satisfy the RIP, we will impose two conditions on the random distribution. First, we require that the distribution will yield a matrix that is norm-preserving, which will require that
Equation:

$$\mathbb{E}\left(\varphi_{ij}^2\right) = \frac{1}{M},$$

and hence the variance of the distribution is $1/M$. Second, we require that the distribution is a *sub-Gaussian* distribution, meaning that there exists a constant $c > 0$ such that
Equation:

$$\mathbb{E}\left(e^{\varphi_{ij}t}\right) \leq e^{c^2t^2/2}$$

for all $t \in \mathbb{R}$. This says that the moment-generating function of our distribution is dominated by that of a Gaussian distribution, which is also equivalent to requiring that tails of our distribution decay *at least as fast* as the tails of a Gaussian distribution. Examples of sub-Gaussian distributions include the Gaussian distribution, the Bernoulli distribution taking values $\pm 1/\sqrt{M}$, and more generally any distribution with bounded support. See "Sub-Gaussian random variables" for more details.

For the moment, we will actually assume a bit more than sub-Gaussianity. Specifically, we will assume that the entries of Φ are *strictly* sub-Gaussian, which means that they satisfy [link] with $c^2 = \mathbb{E}\left(\varphi_{ij}^2\right) = \frac{1}{M}$. Similar results to the following would hold for general sub-Gaussian distributions, but to simplify the constants we restrict our present attention to the strictly sub-Gaussian Φ. In this case we have the following useful result, which is proven in "Concentration of measure for sub-Gaussian random variables".

Suppose that Φ is an $M \times N$ matrix whose entries φ_{ij} are i.i.d. with φ_{ij} drawn according to a strictly sub-Gaussian distribution with $c^2 = 1/M$. Let $Y = \Phi x$ for $x \in \mathbb{R}^N$. Then for any $\epsilon > 0$, and any $x \in \mathbb{R}^N$,
Equation:

$$\mathbb{E}\left(\| Y \|_2^2\right) = \| x \|_2^2$$

and
Equation:

$$\mathbb{P}\left(\left|\| Y \|_2^2 - \| x \|_2^2\right| \geq \epsilon \| x \|_2^2\right) \leq 2 \exp\left(-\frac{M\epsilon^2}{\kappa^*}\right)$$

with $\kappa^* = 2/\left(1 - \log\left(2\right)\right) \approx 6.52$.

This tells us that the norm of a sub-Gaussian random vector strongly concentrates about its mean. Using this result, in "Proof of the RIP for sub-Gaussian matrices" we provide a simple proof based on that in [link] that sub-Gaussian matrices satisfy the RIP.

Fix $\delta \in (0, 1)$. Let Φ be an $M \times N$ random matrix whose entries φ_{ij} are i.i.d. with φ_{ij} drawn according to a strictly sub-Gaussian distribution with $c^2 = 1/M$. If
Equation:

$$M \geq \kappa_1 K \log\left(\frac{N}{K}\right),$$

then Φ satisfies the RIP of order K with the prescribed δ with probability exceeding $1 - 2e^{-\kappa_2 M}$, where κ_1 is arbitrary and
$\kappa_2 = \delta^2/2\kappa^* - \log\left(42e/\delta\right)/\kappa_1$.

Note that in light of the measurement bounds in "The restricted isometry property" we see that [link] achieves the optimal number of measurements (up to a constant).

Using random matrices to construct Φ has a number of additional benefits. To illustrate these, we will focus on the RIP. First, one can show that for random constructions the measurements are *democratic*, meaning that it is possible to recover a signal using any sufficiently large subset of the measurements [link], [link]. Thus, by using random Φ one can be robust to the loss or corruption of a small fraction of the measurements. Second, and perhaps more significantly, in practice we are often more interested in the setting where x is sparse with respect to some basis Ψ. In this case what we actually require is that the product $\Phi\Psi$ satisfies the RIP. If we were to use a

deterministic construction then we would need to explicitly take Ψ into account in our construction of Φ, but when Φ is chosen randomly we can avoid this consideration. For example, if Φ is chosen according to a Gaussian distribution and Ψ is an orthonormal basis then one can easily show that $\Phi\Psi$ will also have a Gaussian distribution, and so provided that M is sufficiently high $\Phi\Psi$ will satisfy the RIP with high probability, just as before. Although less obvious, similar results hold for sub-Gaussian distributions as well [link]. This property, sometimes referred to as *universality*, constitutes a significant advantage of using random matrices to construct Φ.

Finally, we note that since the fully random matrix approach is sometimes impractical to build in hardware, several hardware architectures have been implemented and/or proposed that enable random measurements to be acquired in practical settings. Examples include the random demodulator [link], random filtering [link], the modulated wideband converter [link], random convolution [link], [link], and the compressive multiplexer [link]. These architectures typically use a reduced amount of randomness and are modeled via matrices Φ that have significantly more structure than a fully random matrix. Perhaps somewhat surprisingly, while it is typically not quite as easy as in the fully random case, one can prove that many of these constructions also satisfy the RIP.

Coherence
In this module we introduce coherence, which provides a more computationally friendly alternative to conditions such as the spark, NSP, or RIP. We briefly describe the theoretical relationship between these conditions.

While the spark, null space property (NSP), and restricted isometry property (RIP) all provide guarantees for the recovery of sparse signals, verifying that a general matrix Φ satisfies any of these properties has a combinatorial computational complexity, since in each case one must essentially consider $\binom{N}{K}$ submatrices. In many settings it is preferable to use properties of Φ that are easily computable to provide more concrete recovery guarantees. The *coherence* of a matrix is one such property [link], [link].

> The coherence of a matrix Φ, $\mu(\Phi)$, is the largest absolute inner product between any two columns φ_i, φ_j of Φ:
> **Equation:**

$$\mu(\Phi) = \max_{1 \le i < j \le N} \frac{|\langle \varphi_i, \varphi_j \rangle|}{\| \varphi_i \|_2 \| \varphi_j \|_2}.$$

It is possible to show that the coherence of a matrix is always in the range $\mu(\Phi) \in \left[\sqrt{\frac{N-M}{M(N-1)}}, 1 \right]$; the lower bound is known as the Welch bound [link], [link], [link]. Note that when $N \gg M$, the lower bound is approximately $\mu(\Phi) \geq 1/\sqrt{M}$.

One can sometimes relate coherence to the spark, NSP, and RIP. For example, the coherence and spark properties of a matrix can be related by employing the Gershgorin circle theorem [link], [link].
(Theorem 2 of [link])

The eigenvalues of an $N \times N$ matrix M with entries m_{ij}, $1 \leq i, j \leq N$, lie in the union of N discs $d_i = d_i(c_i, r_i)$, $1 \leq i \leq N$, centered at

$c_i = m_{ii}$ and with radius $r_i = \sum_{j \neq i} |m_{ij}|$.

Applying this theorem on the Gram matrix $G = \Phi_\Lambda^T \Phi_\Lambda$ leads to the following straightforward result.

For any matrix Φ,
Equation:

$$\operatorname{spark}(\Phi) \geq 1 + \frac{1}{\mu(\Phi)}.$$

Since $\operatorname{spark}(\Phi)$ does not depend on the scaling of the columns, we can assume without loss of generality that Φ has unit-norm columns. Let $\Lambda \subseteq \{1, ..., N\}$ with $|\Lambda| = p$ determine a set of indices. We consider the restricted Gram matrix $G = \Phi_\Lambda^T \Phi_\Lambda$, which satisfies the following properties:

- $g_{ii} = 1, 1 \leq i \leq p$;
- $|g_{ij}| \leq \mu(\Phi), 1 \leq i, j \leq p, i \neq j$.

From [link], if $\sum_{j \neq i} \left| g_{ij} \right| < |g_{ii}|$ then the matrix G is positive definite, so that the columns of Φ_Λ are linearly independent. Thus, the spark condition implies $(p-1)\mu(\Phi) < 1$ or, equivalently, $p < 1 + 1/\mu(\Phi)$ for all $p < \operatorname{spark}(\Phi)$, yielding $\operatorname{spark}(\Phi) \geq 1 + 1/\mu(\Phi)$.

By merging Theorem 1 from "Null space conditions" with [link], we can pose the following condition on Φ that guarantees uniqueness.
(Theorem 12 of [link])

If
Equation:

$$K < \frac{1}{2}\left(1 + \frac{1}{\mu(\Phi)}\right),$$

then for each measurement vector $y \in \mathbb{R}^M$ there exists at most one signal $x \in \Sigma_K$ such that $y = \Phi x$.

[link], together with the Welch bound, provides an upper bound on the level of sparsity K that guarantees uniqueness using coherence: $K = O\left(\sqrt{M}\right)$. Another straightforward application of the Gershgorin circle theorem ([link]) connects the RIP to the coherence property.

If Φ has unit-norm columns and coherence $\mu = \mu(\Phi)$, then Φ satisfies the RIP of order K with $\delta = (K-1)\mu$ for all $K < 1/\mu$.

The proof of this lemma is similar to that of [link].

The results given here emphasize the need for small coherence $\mu(\Phi)$ for the matrices used in CS. Coherence bounds have been studied both for deterministic and randomized matrices. For example, there are known matrices Φ of size $M \times M^2$ that achieve the coherence lower bound $\mu(\Phi) = 1/\sqrt{M}$, such as the Gabor frame generated from the Alltop sequence [link] and more general equiangular tight frames [link]. These constructions restrict the number of measurements needed to recover a K-sparse signal to be $M = O\left(K^2 \log N\right)$. Furthermore, it can be shown that when the distribution used has zero mean and finite variance, then in the asymptotic regime (as M and N grow) the coherence converges to $\mu(\Phi) = \sqrt{(2 \log N)/M}$ [link], [link], [link]. Such constructions would allow K to grow asymptotically as $M = O\left(K^2 \log N\right)$, matching the known finite-dimensional bounds.

The measurement bounds dependent on coherence are handicapped by the squared dependence on the sparsity K, but it is possible to overcome this bottleneck by shifting the types of guarantees from worst-case/deterministic behavior, to average-case/probabilistic behavior [link], [link]. In this setting, we pose a probabilistic prior on the set of K-sparse signals $x \in \Sigma_K$. It is then possible to show that if Φ has low coherence $\mu(\Phi)$ and spectral norm $\| \Phi \|_2$, and if $K = O\left(\mu^{-2}(\Phi) \log N\right)$, then the signal x can be recovered from its CS measurements $y = \Phi x$ with high probability. Note that if we replace the Welch bound, then we obtain $K = O(M \log N)$,

which returns to the linear dependence of the measurement bound on the signal sparsity that appears in RIP-based results.

Signal recovery via ℓ_1 minimization
This module introduces and motivates ℓ_1 minimization as a framework for sparse recovery.

As we will see later in this course, there now exist a wide variety of approaches to recover a sparse signal x from a small number of linear measurements. We begin by considering a natural first approach to the problem of sparse recovery.

Given measurements $y = \Phi x$ and the knowledge that our original signal x is sparse or compressible, it is natural to attempt to recover x by solving an optimization problem of the form
Equation:

$$\widehat{x} = \underset{z}{\operatorname{argmin}} \quad \|z\|_0 \quad \text{subject to} \quad z \in \mathscr{B}(y),$$

where $\mathscr{B}(y)$ ensures that $\widehat{x}$ is consistent with the measurements y. Recall that $\|z\|_0 = |\operatorname{supp}(z)|$ simply counts the number of nonzero entries in z, so [link] simply seeks out the sparsest signal consistent with the observed measurements. For example, if our measurements are exact and noise-free, then we can set $\mathscr{B}(y) = \{z : \Phi z = y\}$. When the measurements have been contaminated with a small amount of bounded noise, we could instead set $\mathscr{B}(y) = \{z : \|\Phi z - y\|_2 \leq \epsilon\}$. In both cases, [link] finds the sparsest x that is consistent with the measurements y.

Note that in [link] we are inherently assuming that x itself is sparse. In the more common setting where $x = \Psi\alpha$, we can easily modify the approach and instead consider
Equation:

$$\widehat{\alpha} = \underset{z}{\operatorname{argmin}} \quad \|z\|_0 \quad \text{subject to} \quad z \in \mathscr{B}(y)$$

where $\mathscr{B}(y) = \{z : \Phi\Psi z = y\}$ or $\mathscr{B}(y) = \{z : \|\Phi\Psi z - y\|_2 \leq \epsilon\}$. By setting $\widetilde{\Phi} = \Phi\Psi$ we see that [link] and [link] are essentially identical. Moreover, as noted in "Matrices that satisfy the RIP", in many cases the

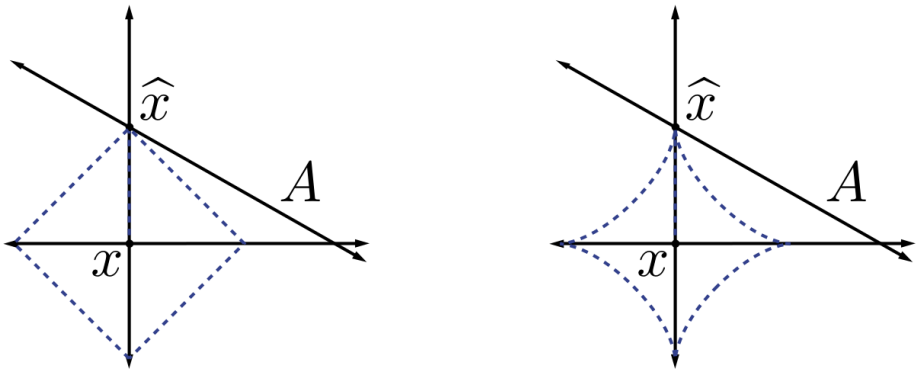
introduction of Ψ does not significantly complicate the construction of matrices Φ such that $\widetilde{\Phi}$ will satisfy the desired properties. Thus, for most of the remainder of this course we will restrict our attention to the case where $\Psi = I$. It is important to note, however, that this restriction does impose certain limits in our analysis when Ψ is a general dictionary and not an orthonormal basis. For example, in this case $\|\widehat{x} - x\|_2 = \|\Psi\hat{c} - \Psi c\|_2 \neq \|\widehat{\alpha} - \alpha\|_2$, and thus a bound on $\|\hat{c} - c\|_2$ cannot directly be translated into a bound on $\|\widehat{x} - x\|$, which is often the metric of interest.

Although it is possible to analyze the performance of [link] under the appropriate assumptions on Φ, we do not pursue this strategy since the objective function $\|\cdot\|_0$ is nonconvex, and hence [link] is potentially very difficult to solve. In fact, one can show that for a general matrix Φ, even finding a solution that approximates the true minimum is NP-hard. One avenue for translating this problem into something more tractable is to replace $\|\cdot\|_0$ with its convex approximation $\|\cdot\|_1$. Specifically, we consider
Equation:

$$\widehat{x} = \underset{z}{\operatorname{argmin}} \quad \|z\|_1 \quad \text{subject to} \quad z \in \mathscr{B}(y).$$

Provided that $\mathscr{B}(y)$ is convex, [link] is computationally feasible. In fact, when $\mathscr{B}(y) = \{z : \Phi z = y\}$, the resulting problem can be posed as a linear program [link].

Approximation in ℓ_1 norm Approximation in ℓ_p quasinorm



Best approximation of a point in $\mathbb{R}^2$ by a a one-dimensional subspace using the ℓ_1 norm and the ℓ_p quasinorm with $p = \frac{1}{2}$.

It is clear that replacing [link] with [link] transforms a computationally intractable problem into a tractable one, but it may not be immediately obvious that the solution to [link] will be at all similar to the solution to [link]. However, there are certainly intuitive reasons to expect that the use of ℓ_1 minimization will indeed promote sparsity. As an example, recall the example we discussed earlier shown in [link]. In this case the solutions to the ℓ_1 minimization problem coincided exactly with the solution to the ℓ_p minimization problem for any $p < 1$, and notably, is sparse. Moreover, the use of ℓ_1 minimization to promote or exploit sparsity has a long history, dating back at least to the work of Beurling on Fourier transform extrapolation from partial observations [link].

Additionally, in a somewhat different context, in 1965 Logan [link] showed that a bandlimited signal can be perfectly recovered in the presence of *arbitrary* corruptions on a small interval. Again, the recovery method consists of searching for the bandlimited signal that is closest to the observed signal in the ℓ_1 norm. This can be viewed as further validation of the intuition gained from [link] — the ℓ_1 norm is well-suited to sparse errors.

Historically, the use of ℓ_1 minimization on large problems finally became practical with the explosion of computing power in the late 1970's and early 1980's. In one of its first applications, it was demonstrated that geophysical

signals consisting of spike trains could be recovered from only the high-frequency components of these signals by exploiting ℓ_1 minimization [link], [link], [link]. Finally, in the 1990's there was renewed interest in these approaches within the signal processing community for the purpose of finding sparse approximations to signals and images when represented in overcomplete dictionaries or unions of bases [link], [link]. Separately, ℓ_1 minimization received significant attention in the statistics literature as a method for variable selection in linear regression, known as the Lasso [link].

Thus, there are a variety of reasons to suspect that ℓ_1 minimization will provide an accurate method for sparse signal recovery. More importantly, this also provides a computationally tractable approach to the sparse signal recovery problem. We now provide an overview of ℓ_1 minimization in both the noise-free and noisy settings from a theoretical perspective. We will then further discuss algorithms for performing ℓ_1 minimization later in this course.

Noise-free signal recovery
This module establishes a simple performance guarantee of L1 minimization for signal recovery with noise-free measurements.

We now begin our analysis of
Equation:

$$\widehat{x} = \underset{z}{\operatorname{argmin}} \quad \|z\|_1 \quad \text{subject to} \quad z \in \mathscr{B}(y).$$

for various specific choices of $\mathscr{B}(y)$. In order to do so, we require the following general result which builds on Lemma 4 from "ℓ_1 minimization proof". The key ideas in this proof follow from [link].

Suppose that Φ satisfies the restricted isometry property (RIP) of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$. Let $x, \widehat{x} \in \mathbb{R}^N$ be given, and define $h = \widehat{x} - x$. Let Λ_0 denote the index set corresponding to the K entries of x with largest magnitude and Λ_1 the index set corresponding to the K entries of $h_{\Lambda_0^c}$ with largest magnitude. Set $\Lambda = \Lambda_0 \cup \Lambda_1$. If $\|\widehat{x}\|_1 \leq \|x\|_1$, then
Equation:

$$\|h\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_1 \frac{|\langle \Phi h_\Lambda, \Phi h \rangle|}{\|h_\Lambda\|_2}.$$

where
Equation:

$$C_0 = 2\frac{1 - \left(1 - \sqrt{2}\right)\delta_{2K}}{1 - \left(1 + \sqrt{2}\right)\delta_{2K}}, \quad C_1 = \frac{2}{1 - \left(1 + \sqrt{2}\right)\delta_{2K}}.$$

We begin by observing that $h = h_\Lambda + h_{\Lambda^c}$, so that from the triangle inequality
Equation:

$$\|h\|_2 \leq \|h_\Lambda\|_2 + \|h_{\Lambda^c}\|_2.$$

We first aim to bound $\|h_{\Lambda^c}\|_2$. From Lemma 3 from "ℓ_1 minimization proof" we have

Equation:

$$\|h_{\Lambda^c}\|_2 = \|\sum_{j\geq 2} h_{\Lambda_j}\|_2 \leq \sum_{j\geq 2} \|h_{\Lambda_j}\|_2 \leq \frac{\|h_{\Lambda_0^c}\|_1}{\sqrt{K}},$$

where the Λ_j are defined as before, i.e., Λ_1 is the index set corresponding to the K largest entries of $h_{\Lambda_0^c}$ (in absolute value), Λ_2 as the index set corresponding to the next K largest entries, and so on.

We now wish to bound $\|h_{\Lambda_0^c}\|_1$. Since $\|x\|_1 \geq \|\widehat{x}\|_1$, by applying the triangle inequality we obtain

Equation:

$$\begin{aligned} \|x\|_1 \geq \|x+h\|_1 &= \|x_{\Lambda_0} + h_{\Lambda_0}\|_1 + \|x_{\Lambda_0^c} + h_{\Lambda_0^c}\|_1 \\ &\geq \|x_{\Lambda_0}\|_1 - \|h_{\Lambda_0}\|_1 + \|h_{\Lambda_0^c}\|_1 - \|x_{\Lambda_0^c}\|_1. \end{aligned}$$

Rearranging and again applying the triangle inequality,

Equation:

$$\begin{aligned} \|h_{\Lambda_0^c}\|_1 &\leq \|x\|_1 - \|x_{\Lambda_0}\|_1 + \|h_{\Lambda_0}\|_1 + \|x_{\Lambda_0^c}\|_1 \\ &\leq \|x - x_{\Lambda_0}\|_1 + \|h_{\Lambda_0}\|_1 + \|x_{\Lambda_0^c}\|_1. \end{aligned}$$

Recalling that $\sigma_K(x)_1 = \|x_{\Lambda_0^c}\|_1 = \|x - x_{\Lambda_0}\|_1$,

Equation:

$$\|h_{\Lambda_0^c}\|_1 \leq \|h_{\Lambda_0}\|_1 + 2\sigma_K(x)_1.$$

Combining this with [link] we obtain
Equation:

$$\|h_{\Lambda^c}\|_2 \leq \frac{\|h_{\Lambda_0}\|_1 + 2\sigma_K(x)_1}{\sqrt{K}} \leq \|h_{\Lambda_0}\|_2 + 2\frac{\sigma_K(x)_1}{\sqrt{K}}$$

where the last inequality follows from standard bounds on ℓ_p norms (Lemma 1 from "The RIP and the NSP"). By observing that $\|h_{\Lambda_0}\|_2 \leq \|h_\Lambda\|_2$ this combines with [link] to yield
Equation:

$$\|h\|_2 \leq 2\|h_\Lambda\|_2 + 2\frac{\sigma_K(x)_1}{\sqrt{K}}.$$

We now turn to establishing a bound for $\|h_\Lambda\|_2$. Combining Lemma 4 from "ℓ_1 minimization proof" with [link] and again applying standard bounds on ℓ_p norms we obtain
Equation:

$$\begin{aligned}
\|h_\Lambda\|_2 &\leq \alpha\frac{\|h_{\Lambda_0^c}\|_1}{\sqrt{K}} + \beta\frac{|\langle \Phi h_\Lambda, \Phi h\rangle|}{\|h_\Lambda\|_2} \\
&\leq \alpha\frac{\|h_{\Lambda_0}\|_1 + 2\sigma_K(x)_1}{\sqrt{K}} + \beta\frac{|\langle \Phi h_\Lambda, \Phi h\rangle|}{\|h_\Lambda\|_2} \\
&\leq \alpha\|h_{\Lambda_0}\|_2 + 2\alpha\frac{\sigma_K(x)_1}{\sqrt{K}} + \beta\frac{|\langle \Phi h_\Lambda, \Phi h\rangle|}{\|h_\Lambda\|_2}.
\end{aligned}$$

Since $\|h_{\Lambda_0}\|_2 \leq \|h_\Lambda\|_2$,
Equation:

$$(1-\alpha)\|h_\Lambda\|_2 \leq 2\alpha\frac{\sigma_K(x)_1}{\sqrt{K}} + \beta\frac{|\langle \Phi h_\Lambda, \Phi h\rangle|}{\|h_\Lambda\|_2}.$$

The assumption that $\delta_{2K} < \sqrt{2} - 1$ ensures that $\alpha < 1$. Dividing by $(1 - \alpha)$ and combining with [link] results in
Equation:

$$\|h\|_2 \leq \left(\frac{4\alpha}{1-\alpha} + 2 \right) \frac{\sigma_K(x)_1}{\sqrt{K}} + \frac{2\beta}{1-\alpha} \frac{|\langle \Phi h_\Lambda, \Phi h \rangle|}{\|h_\Lambda\|_2}.$$

Plugging in for α and β yields the desired constants.

[link] establishes an error bound for the class of ℓ_1 minimization algorithms described by [link] when combined with a measurement matrix Φ satisfying the RIP. In order to obtain specific bounds for concrete examples of $\mathscr{B}(y)$, we must examine how requiring $\widehat{x} \in \mathscr{B}\left(y\right)$ affects $|\langle \Phi h_\Lambda, \Phi h \rangle|$. As an example, in the case of noise-free measurements we obtain the following theorem.
(Theorem 1.1 of [link])

Suppose that Φ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$ and we obtain measurements of the form $y = \Phi x$. Then when $\mathscr{B}(y) = \{z : \Phi z = y\}$, the solution $\widehat{x}$ to [link] obeys
Equation:

$$\|\widehat{x} - x\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}}.$$

Since $x \in \mathscr{B}(y)$ we can apply [link] to obtain that for $h = \widehat{x} - x$,
Equation:

$$\|h\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_1 \frac{|\langle \Phi h_\Lambda, \Phi h \rangle|}{\|h_\Lambda\|_2}.$$

Furthermore, since $x, \widehat{x} \in \mathscr{B}\left(y\right)$ we also have that $y = \Phi x = \Phi \widehat{x}$ and hence $\Phi h = 0$. Therefore the second term vanishes, and we obtain the desired result.

[link] is rather remarkable. By considering the case where $x \in \Sigma_K = \{x : \|x\|_0 \leq K\}$ we can see that provided Φ satisfies the RIP — which as shown earlier allows for as few as $O(K \log (N/K))$ measurements — we can recover any K-sparse x*exactly*. This result seems improbable on its own, and so one might expect that the procedure would be highly sensitive to noise, but we will see next that [link] can also be used to demonstrate that this approach is actually stable.

Note that [link] assumes that Φ satisfies the RIP. One could easily modify the argument to replace this with the assumption that Φ satisfies the null space property (NSP) instead. Specifically, if we are only interested in the noiseless setting, in which case h lies in the null space of Φ, then [link] simplifies and its proof could be broken into two steps: (i) show that if Φ satisfies the RIP then it satisfies the NSP (as shown in "The RIP and the NSP"), and (ii) the NSP implies the simplified version of [link]. This proof directly mirrors that of [link]. Thus, by the same argument as in the proof of [link], it is straightforward to show that if Φ satisfies the NSP then it will obey the same error bound.

Signal recovery in noise
This module establishes a number of results concerning various L1 minimization algorithms designed for sparse signal recovery from noisy measurements. The results in this module apply to both bounded noise as well as Gaussian (or more generally, sub-Gaussian) noise.

The ability to perfectly reconstruct a sparse signal from noise-free measurements represents a promising result. However, in most real-world systems the measurements are likely to be contaminated by some form of noise. For instance, in order to process data in a computer we must be able to represent it using a finite number of bits, and hence the measurements will typically be subject to quantization error. Moreover, systems which are implemented in physical hardware will be subject to a variety of different types of noise depending on the setting.

Perhaps somewhat surprisingly, one can show that it is possible to modify
Equation:

$$\widehat{x} = \underset{z}{\operatorname{argmin}} \ \|z\|_1 \quad \text{subject to} \quad z \in \mathscr{B}(y).$$

to stably recover sparse signals under a variety of common noise models [link], [link], [link]. As might be expected, the restricted isometry property (RIP) is extremely useful in establishing performance guarantees in noise.

In our analysis we will make repeated use of Lemma 1 from "Noise-free signal recovery", so we repeat it here for convenience.

Suppose that Φ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$. Let $x, \widehat{x} \in \mathbb{R}^N$ be given, and define $h = \widehat{x} - x$. Let Λ_0 denote the index set corresponding to the K entries of x with largest magnitude and Λ_1 the index set corresponding to the K entries of $h_{\Lambda_0^c}$ with largest magnitude. Set $\Lambda = \Lambda_0 \cup \Lambda_1$. If $\|\widehat{x}\|_1 \leq \|x\|_1$, then
Equation:

$$\|h\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_1 \frac{|\langle \Phi h_\Lambda, \Phi h \rangle|}{\|h_\Lambda\|_2}.$$

where
Equation:

$$C_0 = 2\frac{1-\left(1-\sqrt{2}\right)\delta_{2K}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}, \quad C_1 = \frac{2}{1-\left(1+\sqrt{2}\right)\delta_{2K}}.$$

Bounded noise

We first provide a bound on the worst-case performance for uniformly bounded noise, as first investigated in [link].
(Theorem 1.2 of [link])

Suppose that Φ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$ and let $y = \Phi x + e$ where $\|e\|_2 \leq \epsilon$. Then when $\mathscr{B}(y) = \{z : \|\Phi z - y\|_2 \leq \epsilon\}$, the solution $\widehat{x}$ to [link] obeys
Equation:

$$\|\widehat{x} - x\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_2\epsilon,$$

where
Equation:

$$C_0 = 2\frac{1-\left(1-\sqrt{2}\right)\delta_{2K}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}, \quad C_2 = 4\frac{\sqrt{1+\delta_{2K}}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}.$$

We are interested in bounding $\|h\|_2 = \|\widehat{x} - x\|_2$. Since $\|e\|_2 \leq \epsilon$, $x \in \mathscr{B}(y)$, and therefore we know that $\|\widehat{x}\|_1 \leq \|x\|_1$. Thus we may apply [link], and it remains to bound $|\langle \Phi h_\Lambda, \Phi h\rangle|$. To do this, we observe that
Equation:

$$\|\Phi h\|_2 = \|\Phi\left(\widehat{x} - x\right)\|_2 = \|\Phi\widehat{x} - y + y - \Phi x\|_2 \leq \|\Phi\widehat{x} - y\|_2 + \|y - \Phi x\|_2 \leq 2\epsilon$$

where the last inequality follows since $x, \widehat{x} \in \mathscr{B}(y)$. Combining this with the RIP and the Cauchy-Schwarz inequality we obtain
Equation:

$$|\langle \Phi h_\Lambda, \Phi h \rangle| \leq \|\Phi h_\Lambda\|_2 \|\Phi h\|_2 \leq 2\epsilon \sqrt{1+\delta_{2K}} \|h_\Lambda\|_2.$$

Thus,
Equation:

$$\|h\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_1 2\epsilon \sqrt{1+\delta_{2K}} = C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_2 \epsilon,$$

completing the proof.

In order to place this result in context, consider how we would recover a sparse vector x if we happened to already know the K locations of the nonzero coefficients, which we denote by Λ_0. This is referred to as the *oracle estimator*. In this case a natural approach is to reconstruct the signal using a simple pseudoinverse:
Equation:

$$\begin{aligned} \widehat{x}_{\Lambda_0} &= \Phi_{\Lambda_0}^\dagger y = \left(\Phi_{\Lambda_0}^T \Phi_{\Lambda_0}\right)^{-1} \Phi_{\Lambda_0}^T y \\ \widehat{x}_{\Lambda_0^c} &= 0. \end{aligned}$$

The implicit assumption in [link] is that Φ_{Λ_0} has full column-rank (and hence we are considering the case where Φ_{Λ_0} is the $M \times K$ matrix with the columns indexed by Λ_0^c removed) so that there is a unique solution to the equation $y = \Phi_{\Lambda_0} x_{\Lambda_0}$. With this choice, the recovery error is given by
Equation:

$$\|\widehat{x} - x\|_2 = \|\left(\Phi_{\Lambda_0}^T \Phi_{\Lambda_0}\right)^{-1} \Phi_{\Lambda_0}^T \left(\Phi x + e\right) - x\|_2 = \|\left(\Phi_{\Lambda_0}^T \Phi_{\Lambda_0}\right)^{-1} \Phi_{\Lambda_0}^T e\|_2.$$

We now consider the worst-case bound for this error. Using standard properties of the singular value decomposition, it is straightforward to show that if Φ satisfies the RIP of order $2K$ (with constant δ_{2K}), then the largest singular value of $\Phi_{\Lambda_0}^\dagger$ lies in the range $\left[1/\sqrt{1+\delta_{2K}}, 1/\sqrt{1-\delta_{2K}}\right]$. Thus, if we consider the worst-case recovery error over all e such that $\|e\|_2 \leq \epsilon$, then the recovery error can be bounded by
Equation:

$$\frac{\epsilon}{\sqrt{1+\delta_{2K}}} \leq \left\|\widehat{x}-x\right\|_2 \leq \frac{\epsilon}{\sqrt{1-\delta_{2K}}}.$$

Therefore, if x is exactly K-sparse, then the guarantee for the pseudoinverse recovery method, which is given *perfect knowledge of the true support of* x, cannot improve upon the bound in [link] by more than a constant value.

We now examine a slightly different noise model. Whereas [link] assumed that the noise norm $\left\|e\right\|_2$ was small, the theorem below analyzes a different recovery algorithm known as the *Dantzig selector* in the case where $\left\|\Phi^T e\right\|_\infty$ is small [link]. We will see below that this will lead to a simple analysis of the performance of this algorithm in Gaussian noise.

Suppose that Φ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2}-1$ and we obtain measurements of the form $y = \Phi x + e$ where $\left\|\Phi^T e\right\|_\infty \leq \lambda$. Then when $\mathscr{B}\left(y\right) = \left\{z : \left\|\Phi^T\left(\Phi z - y\right)\right\|_\infty \leq \lambda\right\}$, the solution $\widehat{x}$ to [link] obeys
Equation:

$$\left\|\widehat{x}-x\right\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_3\sqrt{K}\lambda,$$

where
Equation:

$$C_0 = 2\frac{1-\left(1-\sqrt{2}\right)\delta_{2K}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}, \quad C_3 = \frac{4\sqrt{2}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}.$$

The proof mirrors that of [link]. Since $\left\|\Phi^T e\right\|_\infty \leq \lambda$, we again have that $x \in \mathscr{B}(y)$, so $\left\|\widehat{x}\right\|_1 \leq \left\|x\right\|_1$ and thus [link] applies. We follow a similar approach as in [link] to bound $\left|\langle \Phi h_\Lambda, \Phi h\rangle\right|$. We first note that
Equation:

$$\left\|\Phi^T\Phi h\right\|_\infty \leq \left\|\Phi^T\left(\Phi\widehat{x}-y\right)\right\|_\infty + \left\|\Phi^T\left(y-\Phi x\right)\right\|_\infty \leq 2\lambda$$

where the last inequality again follows since $x, \widehat{x} \in \mathscr{B}(y)$. Next, note that $\Phi h_\Lambda = \Phi_\Lambda h_\Lambda$. Using this we can apply the Cauchy-Schwarz inequality to obtain
Equation:

$$|\langle \Phi h_\Lambda, \Phi h \rangle| = \left|\left\langle h_\Lambda, \Phi_\Lambda^T \Phi h \right\rangle\right| \leq \|h_\Lambda\|_2 \|\Phi_\Lambda^T \Phi h\|_2.$$

Finally, since $\|\Phi^T \Phi h\|_\infty \leq 2\lambda$, we have that every coefficient of $\Phi^T \Phi h$ is at most 2λ, and thus $\|\Phi_\Lambda^T \Phi h\|_2 \leq \sqrt{2K}\,(2\lambda)$. Thus,
Equation:

$$\|h\|_2 \leq C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_1 2\sqrt{2K}\lambda = C_0 \frac{\sigma_K(x)_1}{\sqrt{K}} + C_3 \sqrt{K}\lambda,$$

as desired.

Gaussian noise

Finally, we also examine the performance of these approaches in the presence of Gaussian noise. The case of Gaussian noise was first considered in [link], which examined the performance of ℓ_0 minimization with noisy measurements. We now see that [link] and [link] can be leveraged to provide similar guarantees for ℓ_1 minimization. To simplify our discussion we will restrict our attention to the case where $x \in \Sigma_K = \{x : \|x\|_0 \leq K\}$, so that $\sigma_K(x)_1 = 0$ and the error bounds in [link] and [link] depend only on the noise e.

To begin, suppose that the coefficients of $e \in \mathbb{R}^M$ are i.i.d. according to a Gaussian distribution with mean zero and variance σ^2. Since the Gaussian distribution is itself sub-Gaussian, we can apply results such as Corollary 1 from "Concentration of measure for sub-Gaussian random variables" to show that there exists a constant $c_0 > 0$ such that for any $\epsilon > 0$,
Equation:

$$\mathbb{P}\left(\|e\|_2 \geq (1+\epsilon)\sqrt{M}\sigma\right) \leq \exp\left(-c_0 \epsilon^2 M\right).$$

Applying this result to [link] with $\epsilon = 1$, we obtain the following result for the special case of Gaussian noise.

Suppose that Φ satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$. Furthermore, suppose that $x \in \Sigma_K$ and that we obtain measurements of the form $y = \Phi x + e$ where the entries of e are i.i.d. $\mathcal{N}\left(0, \sigma^2\right)$. Then when $\mathscr{B}\left(y\right) = \left\{z : \left\|\Phi z - y\right\|_2 \leq 2\sqrt{M}\sigma\right\}$, the solution $\widehat{x}$ to [link] obeys

Equation:

$$\left\|\widehat{x} - x\right\|_2 \leq 8 \frac{\sqrt{1 + \delta_{2K}}}{1 - \left(1 + \sqrt{2}\right)\delta_{2K}} \sqrt{M}\sigma$$

with probability at least $1 - \exp\left(-c_0 M\right)$.

We can similarly consider [link] in the context of Gaussian noise. If we assume that the columns of Φ have unit norm, then each coefficient of $\Phi^T e$ is a Gaussian random variable with mean zero and variance σ^2. Using standard tail bounds for the Gaussian distribution (see Theorem 1 from "Sub-Gaussian random variables"), we have that

Equation:

$$\mathbb{P}\left(\left|\left[\Phi^T e\right]_i\right| \geq t\sigma\right) \leq \exp\left(-t^2/2\right)$$

for $i = 1, 2, \ldots, n$. Thus, using the union bound over the bounds for different i, we obtain

Equation:

$$\mathbb{P}\left(\left\|\Phi^T e\right\|_\infty \geq 2\sqrt{\log N}\sigma\right) \leq N \exp\left(-2 \log N\right) = \frac{1}{N}.$$

Applying this to [link], we obtain the following result, which is a simplified version of Theorem 1.1 of [link].

Suppose that Φ has unit-norm columns and satisfies the RIP of order $2K$ with $\delta_{2K} < \sqrt{2} - 1$. Furthermore, suppose that $x \in \Sigma_K$ and that we obtain measurements of the form $y = \Phi x + e$ where the entries of e are i.i.d. $\mathcal{N}\left(0, \sigma^2\right)$. Then when $\mathscr{B}\left(y\right) = \left\{z : \left\|\Phi^T\left(\Phi z - y\right)\right\|_\infty \leq 2\sqrt{\log N}\sigma\right\}$, the solution $\widehat{x}$ to [link] obeys

Equation:

$$\|\widehat{x} - x\|_2 \leq 4\sqrt{2}\frac{\sqrt{1+\delta_{2K}}}{1-\left(1+\sqrt{2}\right)\delta_{2K}}\sqrt{K \log N}\sigma$$

with probability at least $1 - \frac{1}{N}$.

Ignoring the precise constants and the probabilities with which the bounds hold (which we have made no effort to optimize), we observe that if $M = O(K \log N)$ then these results appear to be essentially the same. However, there is a subtle difference. Specifically, if M and N are fixed and we consider the effect of varying K, we can see that [link] yields a bound that is adaptive to this change, providing a stronger guarantee when K is small, whereas the bound in [link] does not improve as K is reduced. Thus, while they provide very similar guarantees, there are certain circumstances where the Dantzig selector is preferable. See [link] for further discussion of the comparative advantages of these approaches.

Instance-optimal guarantees revisited
In this module we demonstrate the difficulty of obtaining instance-optimal guarantees in the L2 norm. We then show that it is much easier to obtain such guarantees in the probabilistic setting.

We now briefly return to the noise-free setting to take a closer look at instance-optimal guarantees for recovering non-sparse signals. To begin, recall that in Theorem 1 from "Noise-free signal recovery" we bounded the ℓ_2-norm of the reconstruction error of
Equation:

$$\widehat{x} = \underset{z}{\operatorname{argmin}} \quad \|z\|_1 \quad \text{subject to} \quad z \in \mathscr{B}(y).$$

as
Equation:

$$\|\widehat{x} - x\|_2 \leq C_0 \sigma_K(x)_1 / \sqrt{K}$$

when $\mathscr{B}(y) = \{z : \Phi z = y\}$. One can generalize this result to measure the reconstruction error using the ℓ_p-norm for any $p \in [1, 2]$. For example, by a slight modification of these arguments, one can also show that $\|\widehat{x} - x\|_1 \leq C_0 \sigma_K(x)_1$ (see [link]). This leads us to ask whether we might replace the bound for the ℓ_2 error with a result of the form $\|\widehat{x} - x\|_2 \leq C \sigma_K(x)_2$. Unfortunately, obtaining such a result requires an unreasonably large number of measurements, as quantified by the following theorem of [link].
(Theorem 5.1 of [link])

Suppose that Φ is an $M \times N$ matrix and that $\Delta : \mathbb{R}^M \to \mathbb{R}^N$ is a recovery algorithm that satisfies
Equation:

$$\|x - \Delta(\Phi x)\|_2 \leq C \sigma_K(x)_2$$

for some $K \geq 1$, then $M > \left(1 - \sqrt{1 - 1/C^2}\right) N$.

We begin by letting $h \in \mathbb{R}^N$ denote any vector in $\mathcal{N}(\Phi)$. We write $h = h_\Lambda + h_{\Lambda^c}$ where Λ is an arbitrary set of indices satisfying $|\Lambda| \leq K$. Set $x = h_{\Lambda^c}$, and note that $\Phi x = \Phi h_{\Lambda^c} = \Phi h - \Phi h_\Lambda = -\Phi h_\Lambda$ since $h \in \mathcal{N}(\Phi)$. Since $h_\Lambda \in \Sigma_K$, [link]

implies that $\Delta(\Phi x) = \Delta(-\Phi h_\Lambda) = -h_\Lambda$. Hence,
$\|x - \Delta(\Phi x)\|_2 = \|h_{\Lambda^c} - (-h_\Lambda)\|_2 = \|h\|_2$. Furthermore, we observe that $\sigma_K(x)_2 \leq \|x\|_2$, since by definition $\sigma_K(x)_2 \leq \|x - \widetilde{x}\|_2$ for all $\widetilde{x} \in \Sigma_K$, including $\widetilde{x} = 0$. Thus $\|h\|_2 \leq C\|h_{\Lambda^c}\|_2$. Since $\|h\|_2^2 = \|h_\Lambda\|_2^2 + \|h_{\Lambda^c}\|_2^2$, this yields
Equation:

$$\|h_\Lambda\|_2^2 = \|h\|_2^2 - \|h_{\Lambda^c}\|_2^2 \leq \|h\|_2^2 - \frac{1}{C^2}\|h\|_2^2 = \left(1 - \frac{1}{C^2}\right)\|h\|_2^2.$$

This must hold for any vector $h \in \mathcal{N}(\Phi)$ and for any set of indices Λ such that $|\Lambda| \leq K$. In particular, let $\{v_i\}_{i=1}^{N-M}$ be an orthonormal basis for $\mathcal{N}(\Phi)$, and define the vectors $\{h_i\}_{i=1}^{N}$ as follows:
Equation:

$$h_j = \sum_{i=1}^{N-M} v_i(j) v_i.$$

We note that $h_j = \sum_{i=1}^{N-M} \langle e_j, v_i \rangle v_i$ where e_j denotes the vector of all zeros except for a 1 in the j-th entry. Thus we see that $h_j = P_{\mathcal{N}} e_j$ where $P_{\mathcal{N}}$ denotes an orthogonal projection onto $\mathcal{N}(\Phi)$. Since $\|P_{\mathcal{N}} e_j\|_2^2 + \|P_{\mathcal{N}}^\perp e_j\|_2^2 = \|e_j\|_2^2 = 1$, we have that $\|h_j\|_2 \leq 1$. Thus, by setting $\Lambda = \{j\}$ for h_j we observe that
Equation:

$$\left|\sum_{i=1}^{N-M} |v_i(j)|^2\right|^2 = |h_j(j)|^2 \leq \left(1 - \frac{1}{C^2}\right)\|h_j\|_2^2 \leq 1 - \frac{1}{C^2}.$$

Summing over $j = 1, 2, ..., N$, we obtain
Equation:

$$N\sqrt{1 - 1/C^2} \geq \sum_{j=1}^{N}\sum_{i=1}^{N-M} |v_i(j)|^2 = \sum_{i=1}^{N-M}\sum_{j=1}^{N} |v_i(j)|^2 = \sum_{i=1}^{N-M} \|v_i\|_2^2 = N - M,$$

and thus $M \geq \left(1 - \sqrt{1 - 1/C^2}\right)N$ as desired.

Thus, if we want a bound of the form [link] that holds for *all* signals x with a constant $C \approx 1$, then regardless of what recovery algorithm we use we will need to take $M \approx N$ measurements. However, in a sense this result is overly pessimistic, and we will now see that the results we just established for signal recovery in noise can actually allow us to overcome this limitation by essentially treating the approximation error as noise.

Towards this end, notice that all the results concerning ℓ_1 minimization stated thus far are deterministic instance-optimal guarantees that apply simultaneously to all x given any matrix that satisfies the restricted isometry property (RIP). This is an important theoretical property, but as noted in "Matrices that satisfy the RIP", in practice it is very difficult to obtain a deterministic guarantee that the matrix Φ satisfies the RIP. In particular, constructions that rely on randomness are only known to satisfy the RIP with high probability. As an example, recall Theorem 1 from "Matrices that satisfy the RIP", which opens the door to slightly weaker results that hold only with high probability.

Fix $\delta \in (0, 1)$. Let Φ be an $M \times N$ random matrix whose entries φ_{ij} are i.i.d. with φ_{ij} drawn according to a strictly sub-Gaussian distribution with $c^2 = 1/M$. If
Equation:

$$M \geq \kappa_1 K \log\left(\frac{N}{K}\right),$$

then Φ satisfies the RIP of order K with the prescribed δ with probability exceeding $1 - 2e^{-\kappa_2 M}$, where κ_1 is arbitrary and $\kappa_2 = \delta^2/2\kappa^* - \log\left(42e/\delta\right)/\kappa_1$.

Even within the class of probabilistic results, there are two distinct flavors. The typical approach is to combine a probabilistic construction of a matrix that will satisfy the RIP with high probability with the previous results in this chapter. This yields a procedure that, with high probability, will satisfy a deterministic guarantee applying to all possible signals x. A weaker kind of result is one that states that given a signal x, we can draw a random matrix Φ and with high probability expect certain performance *for that signal* x. This type of guarantee is sometimes called *instance-optimal in probability*. The distinction is essentially whether or not we need to draw a new random Φ for each signal x. This may be an important distinction in practice, but if we assume for the moment that it is permissible to draw a new matrix Φ for each x, then we can see that [link] may be somewhat pessimistic. In order to establish our main result we will rely on the fact, previously used in "Matrices that satisfy the RIP", that sub-Gaussian matrices preserve the norm of an arbitrary vector with high probability. Specifically, a slight modification of Corollary 1 from "Matrices that

satisfy the RIP" shows that for any $x \in \mathbb{R}^N$, if we choose Φ according to the procedure in [link], then we also have that
Equation:

$$\mathbb{P}\left(\| \Phi x \|_2^2 \geq 2\| x \|_2^2\right) \leq \exp\left(-\kappa_3 M\right)$$

with $\kappa_3 = 4/\kappa^*$. Using this we obtain the following result.

Let $x \in \mathbb{R}^N$ be fixed. Set $\delta_{2K} < \sqrt{2} - 1$ Suppose that Φ is an $M \times N$ sub-Gaussian random matrix with $M \geq \kappa_1 K \log\left(N/K\right)$. Suppose we obtain measurements of the form $y = \Phi x$. Set $\epsilon = 2\sigma_K(x)_2$. Then with probability exceeding
$1 - 2\exp\left(-\kappa_2 M\right) - \exp\left(-\kappa_3 M\right)$, when $\mathscr{B}\left(y\right) = \{z : \|\Phi z - y\|_2 \leq \epsilon\}$, the solution $\widehat{x}$ to [link] obeys
Equation:

$$\|\widehat{x} - x\|_2 \leq \frac{8\sqrt{1+\delta_{2K}} - \left(1+\sqrt{2}\right)\delta_{2K}}{1 - \left(1+\sqrt{2}\right)\delta_{2K}}\sigma_K(x)_2.$$

First we recall that, as noted above, from [link] we have that Φ will satisfy the RIP of order $2K$ with probability at least $1 - 2\exp\left(-\kappa_2 M\right)$. Next, let Λ denote the index set corresponding to the K entries of x with largest magnitude and write $x = x_\Lambda + x_{\Lambda^c}$. Since $x_\Lambda \in \Sigma_K$, we can write $\Phi x = \Phi x_\Lambda + \Phi x_{\Lambda^c} = \Phi x_\Lambda + e$. If Φ is sub-Gaussian then from Lemma 2 from "Sub-Gaussian random variables" we have that Φx_{Λ^c} is also sub-Gaussian, and one can apply [link] to obtain that with probability at least $1 - \exp\left(-\kappa_3 M\right)$, $\|\Phi x_{\Lambda^c}\|_2 \leq 2\|x_{\Lambda^c}\|_2 = 2\sigma_K(x)_2$. Thus, applying the union bound we have that with probability exceeding
$1 - 2\exp\left(-\kappa_2 M\right) - \exp\left(-\kappa_3 M\right)$, we satisfy the necessary conditions to apply Theorem 1 from "Signal recovery in noise" to x_Λ, in which case $\sigma_K(x_\Lambda)_1 = 0$ and hence
Equation:

$$\|\widehat{x} - x_\Lambda\|_2 \leq 2C_2\sigma_K(x)_2.$$

From the triangle inequality we thus obtain
Equation:

$$\|\widehat{x} - x\|_2 = \|\widehat{x} - x_\Lambda + x_\Lambda - x\|_2 \leq \|\widehat{x} - x_\Lambda\|_2 + \|x_\Lambda - x\|_2 \leq (2C_2 + 1)\sigma_K(x)_2$$

which establishes the theorem.

Thus, although it is not possible to achieve a deterministic guarantee of the form in [link] without taking a prohibitively large number of measurements, it *is* possible to show that such performance guarantees can hold with high probability while simultaneously taking far fewer measurements than would be suggested by [link]. Note that the above result applies only to the case where the parameter is selected correctly, which requires some limited knowledge of x, namely $\sigma_K(x)_2$. In practice this limitation can easily be overcome through a parameter selection technique such as cross-validation [link], but there also exist more intricate analyses of ℓ_1 minimization that show it is possible to obtain similar performance without requiring an oracle for parameter selection [link]. Note that [link] can also be generalized to handle other measurement matrices and to the case where x is compressible rather than sparse. Moreover, this proof technique is applicable to a variety of the greedy algorithms described later in this course that do not require knowledge of the noise level to establish similar results [link], [link].

The cross-polytope and phase transitions
In this module we provide an overview of the relationship between L1 minimization and random projections of the cross-polytope.

The analysis of ℓ_1 minimization based on the restricted isometry property (RIP) described in "Signal recovery in noise" allows us to establish a variety of guarantees under different noise settings, but one drawback is that the analysis of how many measurements are actually required for a matrix to satisfy the RIP is relatively loose. An alternative approach to analyzing ℓ_1 minimization algorithms is to examine them from a more geometric perspective. Towards this end, we define the closed ℓ_1 ball, also known as the *cross-polytope*:
Equation:

$$C^N = \left\{x \in \mathbb{R}^N : \|x\|_1 \leq 1\right\}.$$

Note that C^N is the convex hull of $2N$ points $\{p_i\}_{i=1}^{2N}$. Let $\Phi C^N \subseteq \mathbb{R}^M$ denote the convex polytope defined as either the convex hull of $\{\Phi p_i\}_{i=1}^{2N}$ or equivalently as
Equation:

$$\Phi C^N = \left\{y \in \mathbb{R}^M : y = \Phi x, x \in C^N\right\}.$$

For any $x \in \Sigma_K = \{x : \|x\|_0 \leq K\}$, we can associate a K-face of C^N with the support and sign pattern of x. One can show that the number of K-faces of ΦC^N is precisely the number of index sets of size K for which signals supported on them can be recovered by
Equation:

$$\widehat{x} = \underset{z}{\operatorname{argmin}} \ \|z\|_1 \quad \text{subject to} \quad z \in \mathscr{B}(y).$$

with $\mathscr{B}(y) = \{z : \Phi z = y\}$. Thus, ℓ_1 minimization yields the same solution as ℓ_0 minimization for all $x \in \Sigma_K$ if and only if the number of K-faces of ΦC^N is identical to the number of K-faces of C^N. Moreover, by counting

the number of K-faces of ΦC^N, we can quantify exactly what fraction of sparse vectors can be recovered using ℓ_1 minimization with Φ as our sensing matrix. See [link], [link], [link], [link], [link] for more details and [link] for an overview of the implications of this body of work. Note also that by replacing the cross-polytope with certain other polytopes (the simplex and the hypercube), one can apply the same technique to obtain results concerning the recovery of more limited signal classes, such as sparse signals with nonnegative or bounded entries [link].

Given this result, one can then study random matrix constructions from this perspective to obtain probabilistic bounds on the number of K-faces of ΦC^N with Φ is generated at random, such as from a Gaussian distribution. Under the assumption that $K = \rho M$ and $M = \gamma N$, one can obtain asymptotic results as $N \to \infty$. This analysis leads to the *phase transition* phenomenon, where for large problem sizes there are sharp thresholds dictating that the fraction of K-faces preserved will tend to either one or zero with high probability, depending on ρ and γ [link].

These results provide sharp bounds on the minimum number of measurements required in the noiseless setting. In general, these bounds are significantly stronger than the corresponding measurement bounds obtained within the RIP-based framework given in "Noise-free signal recovery", which tend to be extremely loose in terms of the constants involved. However, these sharper bounds also require somewhat more intricate analysis and typically more restrictive assumptions on Φ (such as it being Gaussian). Thus, one of the main strengths of the RIP-based analysis presented in "Noise-free signal recovery" and "Signal recovery in noise" is that it gives results for a broad class of matrices that can also be extended to noisy settings.

Sparse recovery algorithms
This module introduces some of the tradeoffs involved in the design of sparse recovery algorithms.

Given noisy compressive measurements $y = \Phi x + e$ of a signal x, a core problem in compressive sensing (CS) is to recover a sparse signal x from a set of measurements y. Considerable efforts have been directed towards developing algorithms that perform fast, accurate, and stable reconstruction of x from y. As we have seen in previous chapters, a "good" CS matrix Φ typically satisfies certain geometric conditions, such as the restricted isometry property (RIP). Practical algorithms exploit this fact in various ways in order to drive down the number of measurements, enable faster reconstruction, and ensure robustness to both numerical and stochastic errors.

The design of sparse recovery algorithms are guided by various criteria. Some important ones are listed as follows.

- **Minimal number of measurements.** Sparse recovery algorithms must require approximately the same number of measurements (up to a small constant) required for the stable embedding of K-sparse signals.
- **Robustness to measurement noise and model mismatch** Sparse recovery algorithms must be stable with regards to perturbations of the input signal, as well as noise added to the measurements; both types of errors arise naturally in practical systems.
- **Speed.** Sparse recovery algorithms must strive towards expending minimal computational resources, Keeping in mind that a lot of applications in CS deal with very high-dimensional signals.
- **Performance guarantees.** In previous chapters, we have already seen a range of performance guarantees that hold for sparse signal recovery using ℓ_1 minimization. In evaluating other algorithms, we will have the same considerations. For example, we can choose to design algorithms that possess instance-optimal or probabilistic guarantees. We can also choose to focus on algorithm performance for the recovery of exactly K-sparse signals x, or consider performance for the recovery of general signals xs. Alternately, we can also

consider algorithms that are accompanied by performance guarantees in either the noise-free or noisy settings.

A multitude of algorithms satisfying some (or even all) of the above have been proposed in the literature. While it is impossible to describe all of them in this chapter, we refer the interested reader to the DSP resources webpage for a more complete list of recovery algorithms. Broadly speaking, recovery methods tend to fall under three categories: convex optimization-based approaches, greedy methods, and combinatorial techniques. The rest of the chapter discusses several properties and example algorithms of each flavor of CS reconstruction.

Convex optimization-based methods
This module provides an overview of convex optimization approaches to sparse signal recovery.

An important class of sparse recovery algorithms fall under the purview of *convex optimization*. Algorithms in this category seek to optimize a convex function of the unknown variable over a (possibly unbounded) convex subset of .

Setup

Let be a convex *sparsity-promoting* cost function (i.e., is small for sparse .) To recover a sparse signal representation from measurements , we may either solve
Equation:

when there is no noise, or solve
Equation:

when there is noise in the measurements. Here, is a cost function that penalizes the distance between the vectors and . For an appropriate penalty parameter , [link] is equivalent to the *unconstrained* formulation:
Equation:

for some . The parameter may be chosen by trial-and-error, or by statistical techniques such as cross-validation [link].

For convex programming algorithms, the most common choices of and are usually chosen as follows: , the -norm of , and – , the -norm of the error between the observed measurements and the linear projections of the target vector . In statistics, minimizing this subject to is known as the *Lasso* problem. More generally, acts as a regularization term and can be replaced by other, more complex, functions; for example, the desired signal may be piecewise constant, and simultaneously have a sparse representation under a known basis transform . In this case, we may use a mixed regularization term:
Equation:

It might be tempting to use conventional convex optimization packages for the above formulations ([link], [link], and [link]). Nevertheless, the above problems pose two key challenges which are specific to practical problems encountered in CS: (i) real-world applications are invariably large-scale (an image of a resolution of pixels leads to optimization over a million variables, well beyond the reach of any standard optimization software package); (ii) the objective function is nonsmooth, and standard smoothing techniques do not yield very good results. Hence, for these problems, conventional algorithms (typically involving matrix factorizations) are not effective or even applicable. These unique challenges encountered in the context of CS have led to considerable interest in developing improved sparse recovery algorithms in the optimization community.

Linear programming

In the noiseless case, the -minimization problem (obtained by substituting in [link]) can be recast as a linear program (LP) with equality constraints. These can be solved in polynomial time ( ) using standard interior-point methods [link]. This was the first feasible reconstruction algorithm used for CS recovery and has strong theoretical

guarantees, as shown earlier in this course. In the noisy case, the problem can be recast as a second-order cone program (SOCP) with quadratic constraints. Solving LPs and SOCPs is a principal thrust in optimization research; nevertheless, their application in practical CS problems is limited due to the fact that both the signal dimension , and the number of constraints , can be very large in many scenarios. Note that both LPs and SOCPs correspond to the constrained formulations in [link] and [link] and are solved using *first order* interior-point methods.

A newer algorithm called “l1_ls" [link] is based on an interior-point algorithm that uses a preconditioned conjugate gradient (PCG) method to approximately solve linear systems in a truncated-Newton framework. The algorithm exploits the structure of the Hessian to construct their preconditioner; thus, this is a second order method. Computational results show that about a hundred PCG steps are sufficient for obtaining accurate reconstruction. This method has been typically shown to be slower than first-order methods, but could be faster in cases where the true target signal is highly sparse.

Fixed-point continuation

As opposed to solving the constrained formulation, an alternate approach is to solve the unconstrained formulation in [link]. A widely used method for solving -minimization problems of the form
Equation:

for a convex and differentiable , is an iterative procedure based on *shrinkage* (also called soft thresholding; see [link] below). In the context of solving [link] with a quadratic , this method was independently proposed and analyzed in [link], [link], [link], [link], and then further studied or extended in [link], [link], [link], [link], [link], [link]. Shrinkage is a classic method used in wavelet-based image denoising. The shrinkage operator on any scalar component can be defined as follows:
Equation:

This concept can be used effectively to solve [link]. In particular, the basic algorithm can be written as following the fixed-point iteration: for , the coefficient of at the time step is given by

Equation:

where serves as a step-length for gradient descent (which may vary with) and is as specified by the user. It is easy to see that the larger is, the larger the allowable distance between and . For a quadratic penalty term , the gradient can be easily computed as a linear function of ; thus each iteration of [link] essentially boils down to a small number of matrix-vector multiplications.

The simplicity of the iterative approach is quite appealing, both from a computational, as well as a code-design standpoint. Various modifications, enhancements, and generalizations to this approach have been proposed, both to improve the efficiency of the basic iteration in [link], and to extend its applicability to various kinds of [link], [link], [link]. In principle, the basic iteration in [link] would not be practically effective without a continuation (or path-following) strategy [link], [link] in which we choose a gradually decreasing sequence of values for the parameter to guide the intermediate iterates towards the final optimal solution.

This procedure is known as *continuation*; in [link], the performance of an algorithm known as Fixed-Point Continuation (FPC) has been compared favorably with another similar method known as Gradient Projection for Sparse Reconstruction (GPSR) [link] and “l1_ls” [link]. A key aspect to solving the unconstrained optimization problem is the choice of the parameter . As discussed above, for CS recovery, may be chosen by trial

and error; for the noiseless constrained formulation, we may solve the corresponding unconstrained minimization by choosing a large value for .

In the case of recovery from noisy compressive measurements, a commonly used choice for the convex cost function is the square of the norm of the *residual*. Thus we have:
Equation:

For this particular choice of penalty function, [link] reduces to the following iteration:
Equation:

which is run until convergence to a fixed point. The algorithm is detailed in pseudocode form below.

```
Inputs: CS matrix  , signal measurements  ,
parameter sequence
Outputs: Signal estimate
initialize:      ,      ,      .
while ħalting criterion false do
    1.
    2.              {take a gradient step}
    3.                  {perform soft
thresholding}
    4.            {update measurement residual}

end while
return
```

Bregman iteration methods

It turns out that an efficient method to obtain the solution to the constrained optimization problem in [link] can be devised by solving a small number of the unconstrained problems in the form of [link]. These subproblems are commonly referred to as *Bregman iterations*. A simple version can be written as follows:
Equation:

—

The problem in the second step can be solved by the algorithms reviewed above. Bregman iterations were introduced in [link] for constrained total variation minimization problems, and was proved to converge for closed, convex functions . In [link], it is applied to [link] for
and shown to converge in a finite number of steps for any . For moderate , the number of iterations needed is typically lesser than 5. Compared to the alternate approach that solves [link] through directly solving the unconstrained problem in [link] with a very large , Bregman iterations are often more stable and sometimes much faster.

Discussion

All the methods discussed in this section optimize a convex function (usually the -norm) over a convex (possibly unbounded) set. This implies *guaranteed* convergence to the global optimum. In other words, given that the sampling matrix satisfies the conditions specified in "Signal recovery via minimization", convex optimization methods will recover the underlying signal . In addition, convex relaxation methods also guarantee *stable* recovery by reformulating the recovery problem as the SOCP, or the unconstrained formulation.

Greedy algorithms
In this module we provide an overview of some of the most common greedy algorithms and their application to the problem of sparse recovery.

Setup

As opposed to solving a (possibly computationally expensive) convex optimization program, an alternate flavor to sparse recovery is to apply methods of *sparse approximation*. Recall that the goal of sparse recovery is to recover the *sparsest* vector x which explains the linear measurements y. In other words, we aim to solve the (nonconvex) problem:
Equation:

$$\min_{\mathscr{I}} \left\{ |\mathscr{I}| : y = \sum_{i \in \mathscr{I}} \varphi_i x_i \right\},$$

where $\mathscr{I}$ denotes a particular subset of the indices $i = 1, ..., N$, and φ_i denotes the i^{th} column of Φ. It is well known that searching over the power set formed by the columns of Φ for the optimal subset $\mathscr{I}^*$ with smallest cardinality is NP-hard. Instead, classical sparse approximation methods tackle this problem by *greedily* selecting columns of Φ and forming successively better approximations to y.

Matching Pursuit

Matching Pursuit (MP), named and introduced to the signal processing community by Mallat and Zhang [link], [link], is an iterative greedy algorithm that decomposes a signal into a linear combination of elements from a dictionary. In sparse recovery, this dictionary is merely the sampling matrix $\Phi \in \mathbb{R}^{M \times N}$; we seek a sparse representation (x) of our "signal" y.

MP is conceptually very simple. A key quantity in MP is the *residual* $r \in \mathbb{R}^M$; the residual represents the as-yet "unexplained" portion of the measurements. At each iteration of the algorithm, we select a vector from the dictionary that is maximally correlated with the residual r:

Equation:

$$\lambda_k = \underset{\lambda}{\operatorname{argmax}} \frac{\langle r_k, \varphi_\lambda \rangle \varphi_\lambda}{\| \varphi_\lambda \|^2}.$$

Once this column is selected, we possess a “better” representation of the signal, since a new coefficient indexed by λ_k has been added to our signal approximation. Thus, we update both the residual and the approximation as follows:

Equation:

$$\begin{aligned} r_k &= r_{k-1} - \frac{\langle r_{k-1}, \varphi_{\lambda_k} \rangle \varphi_{\lambda_k}}{\| \varphi_{\lambda_k} \|^2}, \\ \widehat{x}_{\lambda_k} &= \widehat{x}_{\lambda_k} + \langle r_{k-1}, \varphi_{\lambda_k} \rangle. \end{aligned}$$

and repeat the iteration. A suitable stopping criterion is when the norm of r becomes smaller than some quantity. MP is described in pseudocode form below.

```
Inputs: Measurement matrix Φ, signal measurements
y
Outputs: Sparse signal x̂
initialize: x̂_0 = 0, r = y, i = 0.
while ħalting criterion false do
    1. i ← i + 1
    2. b ← Φ^T r {form residual signal estimate}
    3. x̂_i ← x̂_{i−1} + T(1) {update largest magnitude
coefficient}
    4. r ← r − Φx̂_i {update measurement residual}
end while
return x̂ ← x̂_i
```

Although MP is intuitive and can find an accurate approximation of the signal, it possesses two major drawbacks: (i) it offers no guarantees in terms

of recovery error; indeed, it does not exploit the special structure present in the dictionary Φ; (ii) the required number of iterations required can be quite large. The complexity of MP is $O(MNT)$ [link] , where T is the number of MP iterations

Orthogonal Matching Pursuit (OMP)

Matching Pursuit (MP) can prove to be computationally infeasible for many problems, since the complexity of MP grows linearly in the number of iterations T. By employing a simple modification of MP, the maximum number of MP iterations can be upper bounded as follows. At any iteration k, Instead of subtracting the contribution of the dictionary element with which the residual r is maximally correlated, we compute the projection of r onto the *orthogonal subspace* to the linear span of the currently selected dictionary elements. This quantity thus better represents the "unexplained" portion of the residual, and is subtracted from r to form a new residual, and the process is repeated. If Φ_Ω is the submatrix formed by the columns of Φ selected at time step t, the following operations are performed:
Equation:

$$\begin{aligned} x_k &= \operatorname{argmin}_x \| y - \Phi_\Omega x \|_2, \\ \widehat{\alpha}_t &= \Phi_\Omega x_t, \\ r_t &= y - \widehat{\alpha}_t. \end{aligned}$$

These steps are repeated until convergence. This is known as Orthogonal Matching Pursuit (OMP) [link]. Tropp and Gilbert [link] proved that OMP can be used to recover a sparse signal with high probability using compressive measurements. The algorithm converges in at most K iterations, where K is the sparsity, but requires the added computational cost of orthogonalization at each iteration. Indeed, the total complexity of OMP can be shown to be $O(MNK)$.

While OMP is provably fast and can be shown to lead to exact recovery, the guarantees accompanying OMP for sparse recovery are weaker than those associated with optimization techniques. In particular, the reconstruction

guarantees are *not uniform*, i.e., it cannot be shown that a single measurement matrix with $M = CK \log N$ rows can be used to recover every possible $K-$sparse signal with $M = CK \log N$ measurements. (Although it is possible to obtain such uniform guarantees when it is acceptable to take more measurements. For example, see [link].) Another issue with OMP is robustness to noise; it is unknown whether the solution obtained by OMP will only be perturbed slightly by the addition of a small amount of noise in the measurements. Nevertheless, OMP is an efficient method for CS recovery, especially when the signal sparsity K is low. A pseudocode representation of OMP is shown below.

Inputs: Measurement matrix Φ, signal measurements y
Outputs: Sparse representation $\widehat{x}$
Initialize: $\hat{\theta}_0 = 0,\ r = y,\ \Omega = \emptyset,\ i = 0.$
while halting criterion false **do**
    1. $i \leftarrow i + 1$
    2. $b \leftarrow \Phi^T r$ {form residual signal estimate}
    3. $\Omega \leftarrow \Omega \cup \text{supp}(\mathbf{T}(b, 1))$ {add index of residual's largest magnitude entry to signal support}
    4. $\widehat{x}_i\big|_{\Omega} \leftarrow \Phi_{\Omega}^{\dagger} x,\ \widehat{x}_i\big|_{\Omega^C} \leftarrow 0$ {form signal estimate}
    5. $r \leftarrow y - \Phi \widehat{x}_i$ {update measurement residual}
end while
return $\widehat{x} \leftarrow \widehat{x}_i$

Stagewise Orthogonal Matching Pursuit (StOMP)

Orthogonal Matching Pursuit is ineffective when the signal is not very sparse as the computational cost increases quadratically with the number of nonzeros K. In this setting, Stagewise Orthogonal Matching Pursuit (StOMP) [link] is a better choice for approximately sparse signals in a large-scale setting.

StOMP offers considerable computational advantages over ℓ_1 minimization and Orthogonal Matching Pursuit for large scale problems with sparse solutions. The algorithm starts with an initial residual $r_0 = y$ and calculates the set of all projections $\Phi^T r_{k-1}$ at the k^{th} stage (as in OMP). However, instead of picking a single dictionary element, it uses a threshold parameter τ to determine the next best *set of columns* of Φ whose correlations with the current residual exceed τ. The new residual is calculated using a least squares estimate of the signal using this expanded set of columns, just as before.

Unlike OMP, the number of iterations in StOMP is fixed and chosen before hand; $S = 10$ is recommended in [link]. In general, the complexity of StOMP is $O(KN \log N)$, a significant improvement over OMP. However, StOMP does not bring in its wake any reconstruction guarantees. StOMP also has moderate memory requirements compared to OMP where the orthogonalization requires the maintenance of a Cholesky factorization of the dictionary elements.

Compressive Sampling Matching Pursuit (CoSaMP)

Greedy pursuit algorithms (such as MP and OMP) alleviate the issue of computational complexity encountered in optimization-based sparse recovery, but lose the associated strong guarantees for uniform signal recovery, given a requisite number of measurements of the signal. In addition, it is unknown whether these greedy algorithms are robust to signal and/or measurement noise.

There have been some recent attempts to develop greedy algorithms (Regularized OMP [link], [link], Compressive Sampling Matching Pursuit (CoSaMP) [link] and Subspace Pursuit [link]) that bridge this gap between uniformity and complexity. Intriguingly, the restricted isometry property (RIP), developed in the context of analyzing ℓ_1 minimization, plays a central role in such algorithms. Indeed, if the matrix Φ satisfies the RIP of order K, this implies that every subset of K columns of the matrix is approximately orthonormal. This property is used to prove strong convergence results of these greedy-like methods.

One variant of such an approach is employed by the CoSaMP algorithm. An interesting feature of CoSaMP is that unlike MP, OMP and StOMP, new indices in a signal estimate can be added *as well as deleted* from the current set of chosen indices. In contrast, greedy pursuit algorithms suffer from the fact that a chosen index (or equivalently, a chosen atom from the dictionary Φ remains in the signal representation until the end. A brief description of CoSaMP is as follows: at the start of a given iteration i, suppose the signal estimate is $\widehat{x}_{i-1}$.

- Form signal residual estimate: $e \leftarrow \Phi^T r$
- Find the biggest $2K$ coefficients of the signal residual e; call this set of indices Ω.
- Merge supports: $T \leftarrow \Omega \cup \operatorname{supp}\left(\widehat{x}_{i-1}\right)$.
- Form signal estimate b by subspace projection: $b|_T \leftarrow \Phi_T^\dagger y$, $b|_{T^C} \leftarrow 0$.
- Prune b by retaining its K largest coefficients. Call this new estimate $\widehat{x}_i$.
- Update measurement residual: $r \leftarrow y - \Phi \widehat{x}_i$.

This procedure is summarized in pseudocode form below.

Inputs: Measurement matrix Φ, measurements y, signal sparsity K
Output: K-sparse approximation $\widehat{x}$ to true signal representation x
Initialize: $\widehat{x}_0 = 0$, $r = y$; $i = 0$
while halting criterion false **do**
    1. $i \leftarrow i + 1$
    2. $e \leftarrow \Phi^T r$ {form signal residual estimate}
    3. $\Omega \leftarrow \operatorname{supp}(\mathbf{T}(e, 2K))$ {prune signal residual estimate}
    4. $T \leftarrow \Omega \cup \operatorname{supp}\left(\widehat{x}_{i-1}\right)$ {merge supports}
    5. $b|_T \leftarrow \Phi_T^\dagger y$, $b|_{T^C}$ {form signal estimate}
    6. $\widehat{x}_i \leftarrow \mathbf{T}\left(b, K\right)$ {prune signal estimate}
    7. $r \leftarrow y - \Phi \widehat{x}_i$ {update measurement residual}
end while

```
end while
return x̂ ← x̂_i
```

As discussed in [link], the key computational issues for CoSaMP are the formation of the signal residual, and the method used for subspace projection in the signal estimation step. Under certain general assumptions, the computational cost of CoSaMP can be shown to be $O(MN)$, which is *independent* of the sparsity of the original signal. This represents an improvement over both greedy algorithms as well as convex methods.

While CoSaMP arguably represents the state of the art in sparse recovery algorithm performance, it possesses one drawback: the algorithm requires prior knowledge of the sparsity K of the target signal. An incorrect choice of input sparsity may lead to a worse guarantee than the actual error incurred by a weaker algorithm such as OMP. The stability bounds accompanying CoSaMP ensure that the error due to an incorrect parameter choice is bounded, but it is not yet known how these bounds translate into practice.

Iterative Hard Thresholding

Iterative Hard Thresholding (IHT) is a well-known algorithm for solving nonlinear inverse problems. The structure of IHT is simple: starting with an initial estimate $\widehat{x}_0$, iterative hard thresholding (IHT) obtains a sequence of estimates using the iteration:
Equation:

$$\widehat{x}_{i+1} = \mathbf{T}\left(\widehat{x}_i + \Phi^T\left(y - \Phi\widehat{x}_i\right), K\right).$$

In [link], Blumensath and Davies proved that this sequence of iterations converges to a fixed point $\widehat{x}$; further, if the matrix Φ possesses the RIP, they showed that the recovered signal $\widehat{x}$ satisfies an instance-optimality guarantee of the type described earlier. The guarantees (as well as the proof technique) are reminiscent of the ones that are derived in the development of other algorithms such as ROMP and CoSaMP.

Discussion

While convex optimization techniques are powerful methods for computing sparse representations, there are also a variety of greedy/iterative methods for solving such problems. Greedy algorithms rely on iterative approximation of the signal coefficients and support, either by iteratively identifying the support of the signal until a convergence criterion is met, or alternatively by obtaining an improved estimate of the sparse signal at each iteration by accounting for the mismatch to the measured data. Some greedy methods can actually be shown to have performance guarantees that match those obtained for convex optimization approaches. In fact, some of the more sophisticated greedy algorithms are remarkably similar to those used for ℓ_1 minimization described previously. However, the techniques required to prove performance guarantees are substantially different. There also exist iterative techniques for sparse recovery based on message passing schemes for sparse graphical models. In fact, some greedy algorithms (such as those in [link], [link]) can be directly interpreted as message passing methods [link].

Combinatorial algorithms
This module introduces the count-min and count-median sketches as representative examples of combinatorial algorithms for sparse recovery.

In addition to convex optimization and greedy pursuit approaches, there is another important class of sparse recovery algorithms that we will refer to as *combinatorial algorithms*. These algorithms, mostly developed by the theoretical computer science community, in many cases pre-date the compressive sensing literature but are highly relevant to the sparse signal recovery problem.

Setup

The oldest combinatorial algorithms were developed in the context of *group testing* [link], [link], [link]. In the group testing problem, we suppose that there are N total items, of which an unknown subset of K elements are anomalous and need to be identified. For example, we might wish to identify defective products in an industrial setting, or identify a subset of diseased tissue samples in a medical context. In both of these cases the vector x indicates which elements are anomalous, i.e., $x_i \neq 0$ for the K anomalous elements and $x_i = 0$ otherwise. Our goal is to design a collection of tests that allow us to identify the support (and possibly the values of the nonzeros) of x while also minimizing the number of tests performed. In the simplest practical setting these tests are represented by a binary matrix Φ whose entries φ_{ij} are equal to 1 if and only if the j^{th} item is used in the i^{th} test. If the output of the test is linear with respect to the inputs, then the problem of recovering the vector x is essentially the same as the standard sparse recovery problem.

Another application area in which combinatorial algorithms have proven useful is computation on *data streams* [link], [link]. Suppose that x_i represents the number of packets passing through a network router with destination i. Simply storing the vector x is typically infeasible since the total number of possible destinations (represented by a 32-bit IP address) is $N = 2^{32}$. Thus, instead of attempting to store x directly, one can store $y = \Phi x$ where Φ is an $M \times N$ matrix with $M \ll N$. In this context the vector y is often called a *sketch*. Note that in this problem y is computed in

a different manner than in the compressive sensing context. Specifically, in the network traffic example we do not ever observe x_i directly; rather, we observe increments to x_i (when a packet with destination i passes through the router). Thus we construct y iteratively by adding the i^{th} column to y each time we observe an increment to x_i, which we can do since $y = \Phi x$ is linear. When the network traffic is dominated by traffic to a small number of destinations, the vector x is compressible, and thus the problem of recovering x from the sketch Φx is again essentially the same as the sparse recovery problem.

Several combinatorial algorithms for sparse recovery have been developed in the literature. A non-exhaustive list includes Random Fourier Sampling [link], HHS Pursuit [link], and Sparse Sequential Matching Pursuit [link]. We do not provide a full discussion of each of these algorithms; instead, we describe two simple methods that highlight the flavors of combinatorial sparse recovery — *count-min* and *count-median*.

The count-min sketch

Define H as the set of all discrete-valued functions $h : \{1, ..., N\} \to \{1, ..., m\}$. Note that H is a finite set of size m^N. Each function $h \in H$ can be specified by a binary *characteristic matrix* $\varphi(h)$ of size $m \times N$, with each column being a binary vector with exactly one 1 at the location j , where $j = h(i)$. To construct the overall *sampling matrix* Φ, we choose d functions $h_1, ..., h_d$ independently from the uniform distribution defined on H, and vertically concatenate their characteristic matrices. Thus, if $M = md$, Φ is a binary matrix of size $M \times N$ with each column containing exactly d ones.

Now given any signal x, we acquire linear measurements $y = \Phi x$. It is easy to visualize the measurements via the following two properties. First, the coefficients of the measurement vector y are naturally grouped according to the "mother" binary functions $\{h_1, ..., h_d\}$. Second, consider the i^{th} coefficient of the measurement vector y, which corresponds to the mother binary function h. Then, the expression for y_i is simply given by:
Equation:

$$y_i = \sum_{j:h(j)=i} x_j.$$

In other words, for a fixed signal coefficient index j, each measurement y_i as expressed above consists of an observation of x_j corrupted by other signal coefficients mapped to the same i by the function h. Signal recovery essentially consists of estimating the signal values from these “corrupted" observations.

The *count-min* algorithm is useful in the special case where the entries of the original signal are positive. Given measurements y using the sampling matrix Φ as constructed above, the estimate of the j^{th} signal entry is given by:
Equation:

$$\widehat{x}_j = \min_l y_i : h_l(j) = i.$$

Intuitively, this means that the estimate of x_j is formed by simply looking at all measurements that comprise of x_j corrupted by other signal values, and picking the one with the lowest magnitude. Despite the simplicity of this algorithm, it is accompanied by an arguably powerful instance-optimality guarantee: if $d = C \log N$ and $m = 4/\alpha K$, then with high probability, the recovered signal $\widehat{x}$ satisfies:
Equation:

$$\| x - \widehat{x} \|_\infty \leq \alpha / K \cdot \| x - x^* \|_1,$$

where x^* represents the best K-term approximation of x in the ℓ_1 sense.

The count-median sketch

For the general setting when the coefficients of the original signal could be either positive or negative, a similar algorithm known as *count-median* can be used. Instead of picking the minimum of the measurements, we compute

the median of all those measurements that are comprised of a corrupted version of x_j and declare it as the signal coefficient estimate, i.e.,
Equation:

$$\widehat{x}_j = \underset{l}{\text{median}} \;\; y_i : h_l(j) = i.$$

The recovery guarantees for count-median are similar to that for count-min, with a different value of the failure probability constant. An important feature of both count-min and count-median is that they require that the measurements be *perfectly noiseless*, in contrast to optimization/greedy algorithms which can tolerate small amounts of measurement noise.

Summary

Although we ultimately wish to recover a sparse signal from a small number of linear measurements in both of these settings, there are some important differences between such settings and the compressive sensing setting studied in this course. First, in these settings it is natural to assume that the designer of the reconstruction algorithm also has full control over Φ , and is thus free to choose Φ in a manner that reduces the amount of computation required to perform recovery. For example, it is often useful to design Φ so that it has few nonzeros, i.e., the sensing matrix itself is also sparse [link], [link], [link]. In general, most methods involve careful construction of the sensing matrix Φ, which is in contrast with the optimization and greedy methods that work with any matrix satisfying a generic condition such as the restricted isometry property. This additional degree of freedom can lead to significantly faster algorithms [link], [link], [link], [link].

Second, note that the computational complexity of all the convex methods and greedy algorithms described above is always at least linear in N, since in order to recover x we must at least incur the computational cost of reading out all N entries of x. This may be acceptable in many typical compressive sensing applications, but this becomes impractical when N is extremely large, as in the network monitoring example. In this context, one may seek to develop algorithms whose complexity is linear only in the

length of the representation of the signal, i.e., its sparsity K. In this case the algorithm does not return a complete reconstruction of x but instead returns only its K largest elements (and their indices). As surprising as it may seem, such algorithms are indeed possible. See [link], [link] for examples.

Bayesian methods
This module provides an overview of the application of Bayesian methods to compressive sensing and sparse recovery.

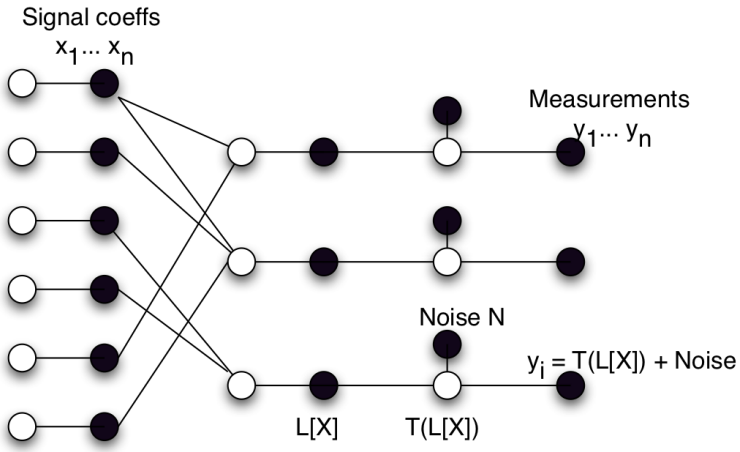
Setup

Throughout this course, we have almost exclusively worked within a deterministic signal framework. In other words, our signal x is fixed and belongs to a known set of signals. In this section, we depart from this framework and assume that the sparse (or compressible) signal of interest arises from a known *probability distribution*, i.e., we assume sparsity promoting *priors* on the elements of x, and recover from the stochastic measurements $y = \Phi x$ a probability distribution on each nonzero element of x. Such an approach falls under the purview of *Bayesian* methods for sparse recovery.

The algorithms discussed in this section demonstrate a digression from the conventional sparse recovery techniques typically used in compressive sensing (CS). We note that none of these algorithms are accompanied by guarantees on the number of measurements required, or the fidelity of signal reconstruction; indeed, in a Bayesian signal modeling framework, there is no well-defined notion of "reconstruction error". However, such methods do provide insight into developing recovery algorithms for rich classes of signals, and may be of considerable practical interest.

Sparse recovery via belief propagation

As we will see later in this course, there are significant parallels to be drawn between error correcting codes and sparse recovery [link]. In particular, sparse codes such as LDPC codes have had grand success. The advantage that sparse coding matrices may have in efficient encoding of signals and their low complexity decoding algorithms, is transferable to CS encoding and decoding with the use of *sparse* sensing matrices Φ. The sparsity in the Φ matrix is equivalent to the sparsity in LDPC coding graphs.



Factor graph depicting the relationship between the variables involved in CS decoding using BP. Variable nodes are black and the constraint nodes are white.

A sensing matrix Φ that defines the relation between the signal x and measurements y can be represented as a bipartite graph of signal coefficient nodes $x(i)$ and measurement nodes $y(i)$ [link], [link]. The factor graph in [link] represents the relationship between the signal coefficients and measurements in the CS decoding problem.

The choice of signal probability density is of practical interest. In many applications, the signals of interest need to be modeled as being compressible (as opposed to being strictly sparse). This behavior is modeled by a two-state Gaussian mixture distribution, with each signal coefficient taking either a “large” or “small” coefficient value state. Assuming that the elements of x are i.i.d., it can be shown that small coefficients occur more frequently than the large coefficients. Other distributions besides the two-state Gaussian may also be used to model the coefficients, for e.g., the i.i.d. Laplace prior on the coefficients of x.

The ultimate goal is to estimate (i.e., decode) x, given y and Φ. The decoding problem takes the form of a Bayesian inference problem in which

we want to approximate the marginal distributions of each of the $x(i)$ coefficients conditioned on the observed measurements $y(i)$. We can then estimate the Maximum Likelihood Estimate (MLE), or the Maximum a Posteriori (MAP) estimates of the coefficients from their distributions. This sort of inference can be solved using a variety of methods; for example, the popular belief propagation method (BP) [link] can be applied to solve for the coefficients approximately. Although exact inference in arbitrary graphical models is an NP hard problem, inference using BP can be employed when Φ is sparse enough, i.e., when most of the entries in the matrix are equal to zero.

Sparse Bayesian learning

Another probabilistic approach used to estimate the components of x is by using Relevance Vector Machines (RVMs). An RVM is essentially a Bayesian learning method that produces sparse classification by linearly weighting a small number of fixed basis functions from a large dictionary of potential candidates (for more details the interested reader may refer to [link], [link]). From the CS perspective, we may view this as a method to determine the elements of a sparse x which linearly weight the basis functions comprising the columns of Φ.

The RVM setup employs a hierarchy of priors; first, a Gaussian prior is assigned to each of the N elements of x; subsequently, a Gamma prior is assigned to the inverse-variance α_i of the i^{th} Gaussian prior. Therefore each α_i controls the strength of the prior on its associated weight in x_i. If x is the sparse vector to be reconstructed, its associated Gaussian prior is given by:
Equation:

$$p\left(x|\alpha\right) = \prod_{i=1}^{N} x_i \ 0, \alpha_i^{-1}$$

and the Gamma prior on α is written as:
Equation:

$$p(\alpha|a,b) = \prod_{i=1}^{N} \Gamma(\alpha_i|a,b)$$

The overall prior on x can be analytically evaluated to be the Student-t distribution, which can be designed to peak at $x_i = 0$ with appropriate choice of a and b. This enables the desired solution x to be sparse. The RVM approach can be visualized using a graphical model similar to the one in "Sparse recovery via belief propagation". Using the observed measurements y, the posterior density on each x_i is estimated by an iterative algorithm (e.g., Markov Chain Monte Carlo (MCMC) methods). For a detailed analysis of the RVM with a measurement noise prior, refer to [link], [link].

Alternatively, we can eliminate the need to set the hyperparameters a and b as follows. Assuming Gaussian measurement noise with mean 0 and variance σ^2, we can directly find the marginal log likelihood for α and maximize it by the EM algorithm (or directly differentiate) to find estimates for α.
Equation:

$$(\alpha) = \log p\ \ y\ \alpha, \sigma^2\ \ = \log \quad p\ \ y\ x, \sigma^2\ \ p(y|\alpha)dx.$$

Bayesian compressive sensing

Unfortunately, evaluation of the log-likelihood in the original RVM setup involves taking the inverse of an $N \times N$ matrix, rendering the algorithm's complexity to be $O\ \ N^3\ \ $. A fast alternative algorithm for the RVM is available which monotonically maximizes the marginal likelihoods of the priors by a gradient ascent, resulting in an algorithm with complexity $O\ \ NM^2\ \ $. Here, basis functions are sequentially added and deleted, thus building the model up constructively, and the true sparsity of the signal x is exploited to minimize model complexity. This is known as Fast Marginal Likelihood Maximization, and is employed by the Bayesian Compressive

Sensing (BCS) algorithm [link] to efficiently evaluate the posterior densities of x_i.

A key advantage of the BCS algorithm is that it enables evaluation of "error bars" on each estimated coefficient of x; these give us an idea of the (in)accuracies of these estimates. These error bars could be used to *adaptively select* the linear projections (i.e., the rows of the matrix Φ) to reduce uncertainty in the signal. This provides an intriguing connection between CS and machine learning techniques such as experimental design and active learning [link], [link].

Linear regression and model selection

This module provides a brief overview of the relationship between model selection, sparse linear regression, and the techniques developed in compressive sensing.

Many of the sparse recovery algorithms we have described so far in this course were originally developed to address the problem of sparse linear regression and model selection in statistics. In this setting we are given some data consisting of a set of input variables and response variables. We will suppose that there are a total of N input variables, and we observe a total of M input and response pairs. We can represent the set of input variable observations as an $M \times N$ matrix Φ, and the set of response variable observations as an $M \times 1$ vector y.

In linear regression, it is assumed that y can be approximated as a linear function of the input variables, i.e., there exists an x such that $y \approx \Phi x$. However, when the number of input variables is large compared to the number of observations, i.e., $M \ll N$, this becomes extremely challenging because we wish to estimate N parameters from far fewer than N observations. In general this would be impossible to overcome, but in practice it is common that only a few input variables are actually necessary to predict the response variable. In this case the x that we wish to estimate is sparse, and we can apply all of the techniques that we have learned so far for sparse recovery to estimate x. In this setting, not only does sparsity aid us in our goal of obtaining a regression, but it also performs *model selection* by identifying the most relevant variables in predicting the response.

Sparse error correction

This module illustrates the application of the ideas of compressive sensing to the design and decoding of error correcting codes for vectors of real numbers subject to sparse corruptions.

In communications, error correction refers to mechanisms that can detect and correct errors in the data that appear duet to distortion in the transmission channel. Standard approaches for error correction rely on repetition schemes, redundancy checks, or nearest neighbor code search. We consider the particular case in which a signal x with M entries is coded by taking length-N linearly independent codewords $\{\varphi_1, ... \varphi_M\}$, with $N > M$ and summing them using the entries of x as coefficients. The received message is a length-N code $y = \sum_{m=1}^{M} \varphi_i x_i = \Phi f$, where Φ is a matrix that has the different codewords for columns. We assume that the transmission channel corrupts the entries of y in an additive way, so that the received data is $y = \Phi x + e$, where e is an error vector.

The techniques developed for sparse recovery in the context of compressive sensing (CS) provide a number of methods to estimate the error vector e — therefore making it possible to correct it and obtain the signal x — when e is sufficiently sparse [link]. To estimate the error, we build a matrix Θ that is a basis for the orthogonal subspace to the span of the matrix Φ, i.e., an $(N - M) \times N$ matrix Θ that holds $\Theta\Phi = 0$. When such a matrix is obtained, we can modify the measurements by multiplying them with the matrix to obtain $\tilde{y} = \Theta y = \Theta\Phi x + \Theta e = \Theta e$. If the matrix Θ is well-suited for CS (i.e., it satisfies a condition such as the restricted isometry property) and e is sufficiently sparse, then the error vector e can be estimated accurately using CS. Once the estimate $\hat{e}$ is obtained, the error-free measurements can be estimated as $\hat{y} = y - \hat{e}$, and the signal can be recovered as $\widehat{x} = \Phi^{\dagger}\hat{y} = \Phi^{\dagger}y - \Phi^{\dagger}\hat{e}$. As an example, when the codewords φ_m have random independent and identically distributed sub-Gaussian entries, then a K-sparse error can be corrected if $M < N - CK \log N/K$ for a fixed constant C (see "Matrices that satisfy the RIP").

Group testing and data stream algorithms
This module provides an overview of the relationship between compressive sensing and problems in theoretical computer science including combinatorial group testing and computation on data streams.

Another scenario where compressive sensing and sparse recovery algorithms can be potentially useful is the context of *group testing* and the related problem of *computation on data streams*.

Group testing

Among the historically oldest of all sparse recovery algorithms were developed in the context of *combinatorial group testing* [link], [link], [link]. In this problem we suppose that there are N total items and K anomalous elements that we wish to find. For example, we might wish to identify defective products in an industrial setting, or identify a subset of diseased tissue samples in a medical context. In both of these cases the vector x indicates which elements are anomalous, i.e., $x_i \neq 0$ for the K anomalous elements and $x_i = 0$ otherwise. Our goal is to design a collection of tests that allow us to identify the support (and possibly the values of the nonzeros) of x while also minimizing the number of tests performed. In the simplest practical setting these tests are represented by a binary matrix Φ whose entries φ_{ij} are equal to 1 if and only if the j^{th} item is used in the i^{th} test. If the output of the test is linear with respect to the inputs, then the problem of recovering the vector x is essentially the same as the standard sparse recovery problem in compressive sensing.

Computation on data streams

Another application area in which ideas related to compressive sensing have proven useful is computation on *data streams* [link], [link]. As an example of a typical data streaming problem, suppose that x_i represents the number of packets passing through a network router with destination i. Simply storing the vector x is typically infeasible since the total number of possible destinations (represented by a 32-bit IP address) is $N = 2^{32}$. Thus, instead of attempting to store x directly, one can store $y = \Phi x$ where Φ is

an $M \times N$ matrix with $M \ll N$. In this context the vector y is often called a *sketch*. Note that in this problem y is computed in a different manner than in the compressive sensing context. Specifically, in the network traffic example we do not ever observe x_i directly, rather we observe increments to x_i (when a packet with destination i passes through the router). Thus we construct y iteratively by adding the i^{th} column to y each time we observe an increment to x_i, which we can do since $y = \Phi x$ is linear. When the network traffic is dominated by traffic to a small number of destinations, the vector x is compressible, and thus the problem of recovering x from the sketch Φx is again essentially the same as the sparse recovery problem in compressive sensing.

Compressive medical imaging
This module describes the application of compressive sensing to problems in medical imaging.

MR image reconstruction

Magnetic Resonance Imaging (MRI) is a medical imaging technique based on the core principle that protons in water molecules in the human body align themselves in a magnetic field. MRI machines repeatedly pulse magnetic fields to cause water molecules in the human body to disorient and then reorient themselves, which causes a release of detectable radiofrequencies. We assume that the object to be imaged as a collection of voxels. The MRI's magnetic pulses are sent incrementally along a gradient leading to a different phase and frequency encoding for each column and row of voxels respectively. Abstracting away from the technicalities of the physical process, the magnetic field measured in MRI acquisition corresponds to a Fourier coefficient of the imaged object; the object can then be recovered by an inverse Fourier transform. , we can view the MRI as measuring Fourier samples.

A major limitation of the MRI process is the linear relation between the number of measured data samples and scan times. Long-duration MRI scans are more susceptible to physiological motion artifacts, add discomfort to the patient, and are expensive [link]. Therefore, minimizing scan time without compromising image quality is of direct benefit to the medical community.

The theory of compressive sensing (CS) can be applied to MR image reconstruction by exploiting the transform-domain sparsity of MR images [link], [link], [link], [link]. In standard MRI reconstruction, undersampling in the Fourier domain results in aliasing artifacts when the image is reconstructed. However, when a known transform renders the object image sparse or compressible, the image can be reconstructed using sparse recovery methods. While the discrete cosine and wavelet transforms are commonly used in CS to reconstruct these images, the use of total variation norm minimization also provides high-quality reconstruction.

Electroencephalography

Electroencephalography (EEG) and Magnetoencephalography (MEG) are two popular noninvasive methods to characterize brain function by measuring scalp electric potential distributions and magnetic fields due to neuronal firing. EEG and MEG provide temporal resolution on the millisecond timescale characteristic of neural population activity and can also help to estimate the current sources inside the brain by solving an inverse problem [link].

Models for neuromagnetic sources suggest that the underlying activity is often limited in spatial extent. Based on this idea, algorithms like FOCUSS (Focal Underdetermined System Solution) are used to identify highly localized sources by assuming a sparse model to solve an underdetermined problem [link].

FOCUSS is a recursive linear estimation procedure, based on a weighted pseudo-inverse solution. The algorithm assigns a current (with nonlinear current location parameters) to each element within a region so that the unknown current values can be related linearly to the measurements. The weights at each step are derived from the solution of the previous iterative step. The algorithm converges to a source distribution in which the number of parameters required to describe source currents does not exceed the number of measurements. The initialization determines which of the localized solutions the algorithm converges to.

Analog-to-information conversion
In this module we describe the random demodulator and how it can be used in the application of the theory of compressive sensing to the problem of acquiring a high-bandwidth continuous-time signal.

We now consider the application of compressive sensing (CS) to the problem of designing a system that can acquire a continuous-time signal $x(t)$. Specifically, we would like to build an *analog-to-digital converter* (ADC) that avoids having to sample $x(t)$ at its Nyquist rate when $x(t)$ is sparse. In this context, we will assume that $x(t)$ has some kind of sparse structure in the Fourier domain, meaning that it is still bandlimited but that much of the spectrum is empty. We will discuss the different possible signal models for mathematically capturing this structure in greater detail below. For now, the challenge is that our measurement system must be built using analog hardware. This imposes severe restrictions on the kinds of operations we can perform.

Analog measurement model

To be more concrete, since we are dealing with a continuous-time signal $x(t)$, we must also consider continuous-time test functions $\{\varphi_j(t)\}_{j=1}^{M}$. We then consider a finite window of time, say $t \in [0, T]$, and would like to collect M measurements of the form
Equation:

$$y[j] = \int_0^T x(t)\varphi_j(t)\,dt.$$

Building an analog system to collect such measurements will require three main components:

1. hardware for generating the test signals $\varphi_j(t)$;
2. M correlators that multiply the signal $x(t)$ with each respective $\varphi_j(t)$;
3. M integrators with a zero-valued initial state.

We could then sample and quantize the output of each of the integrators to collect the measurements $y[j]$. Of course, even in this somewhat idealized setting, it should be clear that what we can build in hardware will constrain our choice of $\varphi_j(t)$ since we cannot reliably and accurately produce (and reproduce) arbitrarily complex $\varphi_j(t)$ in analog hardware. Moreover, the architecture described above requires M correlator/integrator pairs operating in parallel, which will be potentially prohibitively expensive both in dollar cost as well as costs such as size, weight, and power (SWAP).

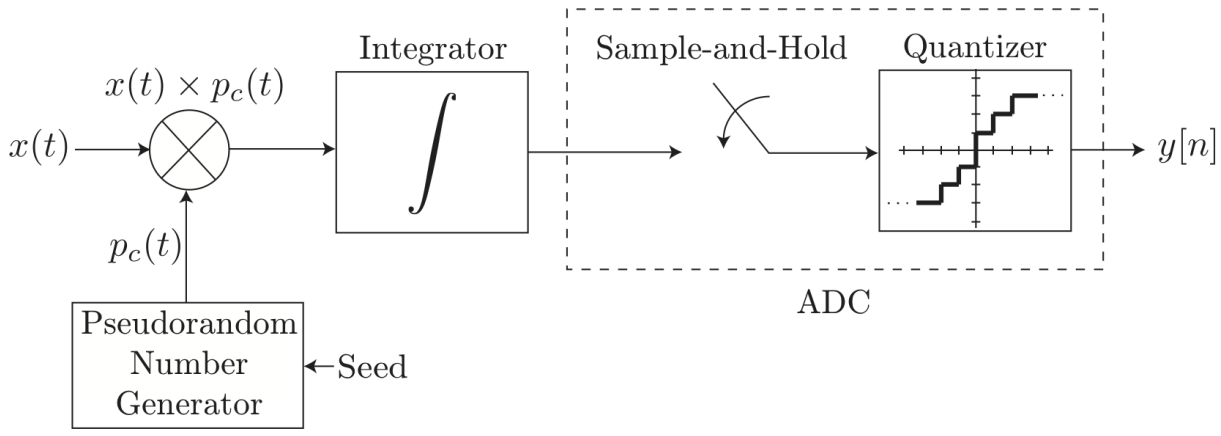
As a result, there have been a number of efforts to design simpler architectures, chiefly by carefully designing structured $\varphi_j(t)$. The simplest to describe and historically earliest idea is to choose $\varphi_j(t) = \delta(t - t_j)$, where $\{t_j\}_{j=1}^{M}$ denotes a sequence of M locations in time at which we would like to sample the signal $x(t)$. Typically, if the number of measurements we are acquiring is lower than the Nyquist-rate, then these locations cannot simply be uniformly spaced in the interval $[0, T]$, but must be carefully chosen. Note that this approach simply requires a single traditional ADC with the ability to sample on a non-uniform grid, avoiding the requirement for M parallel correlator/integrator pairs. Such non-uniform sampling systems have been studied in other contexts outside of the CS framework. For example, there exist specialized fast algorithms for the recovery of extremely large Fourier-sparse signals. The algorithm uses samples at a non-uniform sequence of locations that are highly structured, but where the initial location is chosen using a (pseudo)random seed. This literature provides guarantees similar to those available from standard CS [link], [link]. Additionally, there exist frameworks for the sampling and recovery of multi-band signals, whose Fourier transforms are mostly zero except for a few frequency bands. These schemes again use non-uniform sampling patterns based on coset sampling [link], [link], [link], [link], [link], [link]. Unfortunately, these approaches are often highly sensitive to *jitter*, or error in the timing of when the samples are taken.

We will consider a rather different approach, which we call the *random demodulator* [link], [link], [link].[footnote] The architecture of the random demodulator is depicted in [link]. The analog input $x(t)$ is correlated with a pseudorandom square pulse of ± 1's, called the *chipping sequence* $p_c(t)$,

which alternates between values at a rate of N_aHz, where N_aHz is at least as fast as the Nyquist rate of $x(t)$. The mixed signal is integrated over a time period $1/M_a$ and sampled by a traditional integrate-and-dump back-end ADC at M_aHz $\ll N_a$Hz. In this case our measurements are given by
A correlator is also known as a “demodulator” due to its most common application: demodulating radio signals.
Equation:

$$y[j] = \int_{(j-1)/M_a}^{j/M_a} p_c(t) x(t)\, dt.$$

In practice, data is processed in time blocks of period T, and we define $N = N_a T$ as the number of elements in the chipping sequence, and $M = M_a T$ as the number of measurements. We will discuss the discretization of this model below, but the key observation is that the correlator and chipping sequence operate at a fast rate, while the back-end ADC operates at a low rate. In hardware it is easier to build a high-rate modulator/chipping sequence combination than a high-rate ADC [link]. In fact, many systems already use components of this front end for binary phase shift keying demodulation, as well as for other conventional communication schemes such as CDMA.



Random demodulator block diagram.

Discrete formulation

Although the random demodulator directly acquires compressive measurements without first sampling $x(t)$, it is equivalent to a system which first samples $x(t)$ at its Nyquist-rate to yield a discrete-time vector x, and then applies a matrix Φ to obtain the measurements $y = \Phi x$. To see this we let $p_c[n]$ denote the sequence of ± 1 used to generate the signal $p_c(t)$, i.e., $p_c(t) = p_c[n]$ for $t \in [(n-1)/N_a, n/N_a]$. As an example, consider the first measurement, or the case of $j = 1$. In this case, $t \in [0, 1/M_a]$, so that $p_c(t)$ is determined by $p_c[n]$ for $n = 1, 2, ..., N_a/M_a$. Thus, from [link] we obtain

Equation:

$$\begin{aligned} y[1] &= \int_0^{1/M_a} p_c(t) x(t)\, dt \\ &= \sum_{n=1}^{N_a/M_a} p_c[n] \int_{(n-1)/N_a}^{n/N_a} x(t)\, dt. \end{aligned}$$

But since N_a is the Nyquist-rate of $x(t)$, $\int_{(n-1)/N_a}^{n/N_a} x(t)\, dt$ simply calculates the average value of $x(t)$ on the n^{th} interval, yielding a sample denoted $x[n]$. Thus, we obtain

Equation:

$$y[1] = \sum_{n=1}^{N_a/M_a} p_c[n] x[n].$$

In general, our measurement process is equivalent to multiplying the signal x with the random sequence of ± 1's in $p_c[n]$ and then summing every sequential block of N_a/M_a coefficients. We can represent this as a banded

matrix Φ containing N_a/M_a pseudorandom ± 1s per row. For example, with $N = 12$, $M = 4$, and $T = 1$, such a Φ is expressed as
Equation:

$$\Phi = \begin{matrix} -1 & +1 & +1 & & & & & & & & & \\ & & & -1 & +1 & -1 & & & & & & \\ & & & & & & +1 & +1 & -1 & & & \\ & & & & & & & & & +1 & -1 & -1 \end{matrix}.$$

In general, Φ will have M rows and each row will contain N/M nonzeros. Note that matrices satisfying this structure are extremely efficient to apply, requiring only $O(N)$ computations compared to $O(MN)$ in the general case. This is extremely useful during recovery.

A detailed analysis of the random demodulator in [link] studied the properties of these matrices applied to a particular signal model. Specifically, it is shown that if Ψ represents the $N \times N$ normalized discrete Fourier transform (DFT) matrix, then the matrix $\Phi\Psi$ will satisfy the restricted isometry property (RIP) with high probability, provided that
Equation:

$$M = O\left(K \log^2\left(N/K\right)\right),$$

where the probability is taken with respect to the random choice of $p_c[n]$. This means that if $x(t)$ is a periodic (or finite-length) signal such that once it is sampled it is sparse or compressible in the basis Ψ, then it should be possible to recover $x(t)$ from the measurements provided by the random demodulator. Moreover, it is empirically demonstrated that combining ℓ_1 minimization with the random demodulator can recover K-sparse (in Ψ) signals with
Equation:

$$M \geq CK \log\left(N/K + 1\right)$$

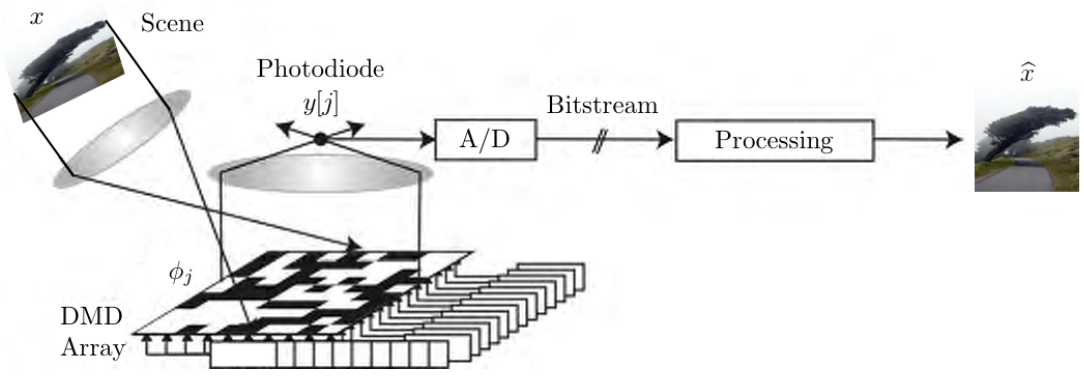
measurements where $C \approx 1.7$ [link].

Note that the signal model considered in [link] is somewhat restrictive, since even a pure tone will not yield a sparse DFT unless the frequency happens to be equal to k/N_a for some integer k. Perhaps a more realistic signal model is the multi-band signal model of [link], [link], [link], [link], [link], [link], where the signal is assumed to be bandlimited outside of K bands each of bandwidth B, where KB is much less than the total possible bandwidth. It remains unknown whether the random demodulator can be exploited to recover such signals. Moreover, there also exist other CS-inspired architectures that we have not explored in this [link], [link], [link], and this remains an active area of research. We have simply provided an overview of one of the more promising approaches in order to illustrate the potential applicability of the ideas of this course to the problem of analog-to-digital conversion.

Single-pixel camera
This module describes the application of compressive sensing to the design of a novel imaging architecture called the "single-pixel camera".

Architecture

Several hardware architectures have been proposed that apply the theory of compressive sensing (CS) in an imaging setting [link], [link], [link]. We will focus on the so-called *single-pixel camera* [link], [link], [link], [link], [link]. The single-pixel camera is an optical computer that sequentially measures the inner products $y[j] = \langle x, \varphi_j \rangle$ between an N-pixel sampled version of the incident light-field from the scene under view (denoted by x) and a set of N-pixel test functions $\{\varphi_j\}_{j=1}^{M}$. The architecture is illustrated in [link], and an aerial view of the camera in the lab is shown in [link]. As shown in these figures, the light-field is focused by a lens (Lens 1 in [link]) not onto a CCD or CMOS sampling array but rather onto a spatial light modulator (SLM). An SLM modulates the intensity of a light beam according to a control signal. A simple example of a transmissive SLM that either passes or blocks parts of the beam is an overhead transparency. Another example is a liquid crystal display (LCD) projector.

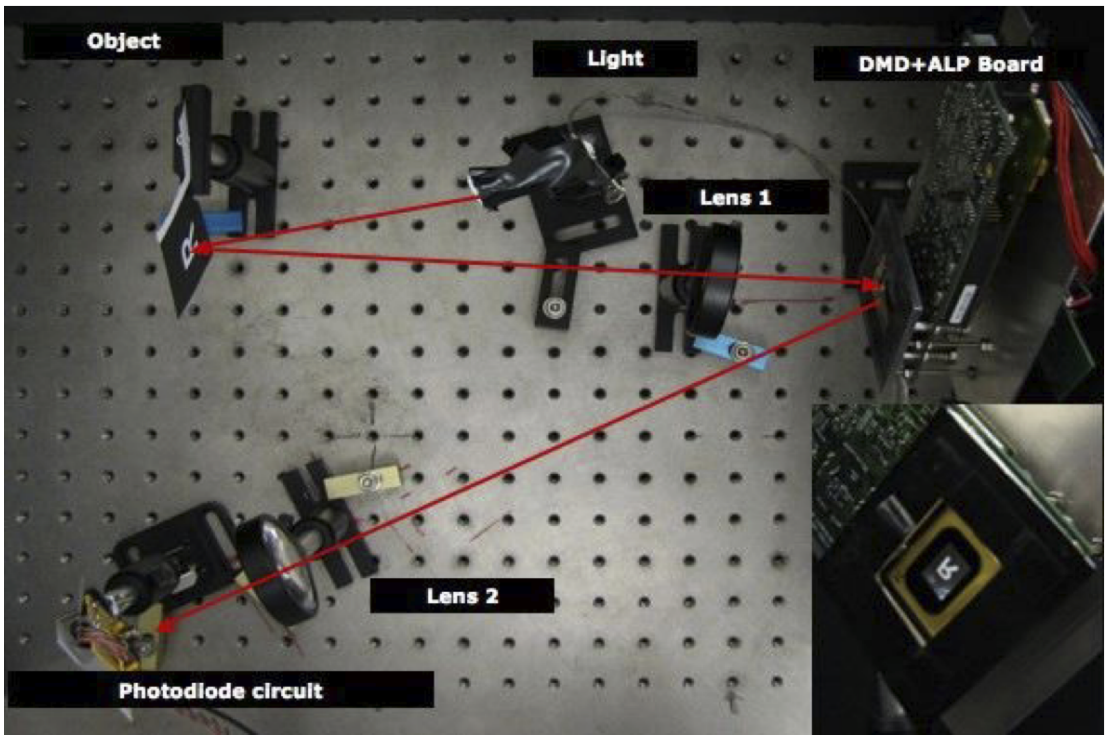


Single-pixel camera block diagram. Incident light-field (corresponding to the desired image x) is reflected off a digital micromirror device (DMD) array whose mirror orientations are modulated according to the pseudorandom pattern φ_j supplied

by a random number generator. Each different mirror pattern produces a voltage at the single photodiode that corresponds to one measurement $y[j]$.

The Texas Instruments (TI) digital micromirror device (DMD) is a reflective SLM that selectively redirects parts of the light beam. The DMD consists of an array of bacterium-sized, electrostatically actuated micro-mirrors, where each mirror in the array is suspended above an individual static random access memory (SRAM) cell. Each mirror rotates about a hinge and can be positioned in one of two states (± 10 degrees from horizontal) according to which bit is loaded into the SRAM cell; thus light falling on the DMD can be reflected in two directions depending on the orientation of the mirrors.

Each element of the SLM corresponds to a particular element of φ_j (and its corresponding pixel in x). For a given φ_j, we can orient the corresponding element of the SLM either towards (corresponding to a 1 at that element of φ_j) or away from (corresponding to a 0 at that element of φ_j) a second lens (Lens 2 in [link]). This second lens collects the reflected light and focuses it onto a single photon detector (the single pixel) that integrates the product of x and φ_j to compute the measurement $y\left[j\right] = \langle x, \varphi_j \rangle$ as its output voltage. This voltage is then digitized by an A/D converter. Values of φ_j between 0 and 1 can be obtained by dithering the mirrors back and forth during the photodiode integration time. By reshaping x into a column vector and the φ_j into row vectors, we can thus model this system as computing the product $y = \Phi x$, where each row of Φ corresponds to a φ_j. To compute randomized measurements, we set the mirror orientations φ_j randomly using a pseudorandom number generator, measure $y[j]$, and then repeat the process M times to obtain the measurement vector y.



Aerial view of the single-pixel camera in the lab.

The single-pixel design reduces the required size, complexity, and cost of the photon detector array down to a single unit, which enables the use of exotic detectors that would be impossible in a conventional digital camera. Example detectors include a photomultiplier tube or an avalanche photodiode for low-light (photon-limited) imaging, a sandwich of several photodiodes sensitive to different light wavelengths for multimodal sensing, a spectrometer for hyperspectral imaging, and so on.

In addition to sensing flexibility, the practical advantages of the single-pixel design include the facts that the quantum efficiency of a photodiode is higher than that of the pixel sensors in a typical CCD or CMOS array and that the fill factor of a DMD can reach 90% whereas that of a CCD/CMOS array is only about 50%. An important advantage to highlight is that each CS measurement receives about $N/2$ times more photons than an average pixel sensor, which significantly reduces image distortion from dark noise and read-out noise.

The single-pixel design falls into the class of multiplex cameras. The baseline standard for multiplexing is classical raster scanning, where the test functions $\{\varphi_j\}$ are a sequence of delta functions $\delta[n-j]$ that turn on each mirror in turn. There are substantial advantages to operating in a CS rather than raster scan mode, including fewer total measurements (M for CS rather than N for raster scan) and significantly reduced dark noise. See [link] for a more detailed discussion of these issues.

[link] (a) and (b) illustrates a target object (a black-and-white printout of an "R") x and reconstructed image $\widehat{x}$ taken by the single-pixel camera prototype in [link] using $N = 256 \times 256$ and $M = N/50$[link]. [link](c) illustrates an $N = 256 \times 256$ color single-pixel photograph of a printout of the Mandrill test image taken under low-light conditions using RGB color filters and a photomultiplier tube with $M = N/10$. In both cases, the images were reconstructed using total variation minimization, which is closely related to wavelet coefficient ℓ_1 minimization [link].

Sample image reconstructions from single-pixel camera. (a) 256×256 conventional image of a black-and-white "R". (b) Image reconstructed from $M = 1300$ single-pixel camera measurements ($50\times$ sub-Nyquist). (c) 256×256 pixel color reconstruction of a printout of the Mandrill test image imaged in a low-light setting using a single photomultiplier tube sensor, RGB color filters, and $M = 6500$ random measurements.

Discrete formulation

Since the DMD array is programmable, we can employ arbitrary test functions φ_j. However, even when we restrict the φ_j to be $\{0, 1\}$-valued, storing these patterns for large values of N is impractical. Furthermore, as noted above, even pseudorandom Φ can be computationally problematic during recovery. Thus, rather than purely random Φ, we can also consider Φ that admit a fast transform-based implementation by taking random submatrices of a Walsh, Hadamard, or noiselet transform [link], [link]. We will describe the Walsh transform for the purpose of illustration.

We will suppose that N is a power of 2 and let $W_{\log_2 N}$ denote the $N \times N$ Walsh transform matrix. We begin by setting $W_0 = 1$, and we now define W_j recursively as

Equation:

$$W_j = \frac{1}{\sqrt{2}} \begin{matrix} W_{j-1} & W_{j-1} \\ W_{j-1} & -W_{j-1} \end{matrix} .$$

This construction produces an orthonormal matrix with entries of $\pm 1/\sqrt{N}$ that admits a fast implementation requiring $O(N \log N)$ computations to apply. As an example, note that

Equation:

$$W_1 = \frac{1}{\sqrt{2}} \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix}$$

and

Equation:

$$W_2 = \frac{1}{2}\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{matrix}.$$

We can exploit these constructions as follows. Suppose that $N = 2^B$ and generate W_B. Let I_Γ denote a $M \times N$ submatrix of the identity I obtained by picking a random set of M rows, so that $I_\Gamma W_B$ is the submatrix of W_B consisting of the rows of W_B indexed by Γ. Furthermore, let D denote a random $N \times N$ permutation matrix. We can generate Φ as
Equation:

$$\Phi = \frac{1}{2}\sqrt{N} I_\Gamma W_B + \frac{1}{2} \quad D.$$

Note that $\frac{1}{2}\sqrt{N} I_\Gamma W_B + \frac{1}{2}$ merely rescales and shifts $I_\Gamma W_B$ to have $\{0, 1\}$-valued entries, and recall that each row of Φ will be reshaped into a 2-D matrix of numbers that is then displayed on the DMD array. Furthermore, D can be thought of as either permuting the pixels or permuting the columns of W_B. This step adds some additional randomness since some of the rows of the Walsh matrix are highly correlated with coarse scale wavelet basis functions — but permuting the pixels eliminates this structure. Note that at this point we do not have any strict guarantees that such Φ combined with a wavelet basis Ψ will yield a product $\Phi\Psi$ satisfying the restricted isometry property, but this approach seems to work well in practice.

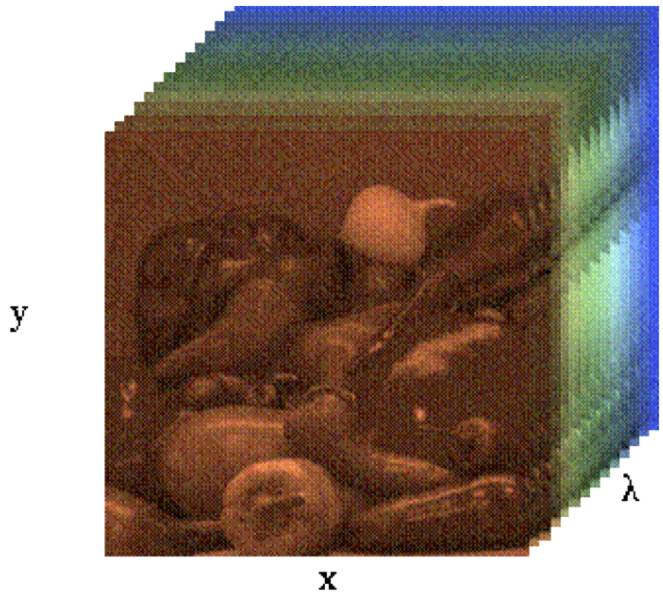
Hyperspectral imaging

This module provides an overview of architectures and methods for hyperspectral imaging using the ideas of compressive sensing.

Standard digital color images of a scene of interest consist of three components – red, green and blue – which contain the intensity level for each of the pixels in three different groups of wavelengths. This concept has been extended in the *hyperspectral* and *multispectral* imaging sensing modalities, where the data to be acquired consists of a three-dimensional *datacube* that has two spatial dimensions x and y and one spectral dimension λ.

In simple terms, a datacube is a 3-D function $f(x, y, \lambda)$ that can be represented as a stacking of intensities of the scene at different wavelengths. An example datacube is shown in [link]. Each of its entries is called a voxel. We also define a pixel's *spectral signature* as the stacking of its voxels in the spectral dimension $f(x, y) = \{f(x, y, \lambda)\}_{\lambda}$. The spectral signature of a pixel can give a wealth of information about the corresponding point in the scene that is not captured by its color. For example, using spectral signatures, it is possible to identify the type of material observed (for example, vegetation vs. ground vs. water), or its chemical composition.

Datacubes are high-dimensional, since the standard number of pixels present in a digitized image is multiplied by the number of spectral bands desired. However, considerable structure is present in the observed data. The spatial structure common in natural images is also observed in hyperspectral imaging, while each pixel's spectral signature is usually smooth.



Example hyperspectral datacube, with labeled dimensions.

Compressive sensing (CS) architectures for hyperspectral imaging perform lower-dimensional projections that multiplex in the spatial domain, the spectral domain, or both. Below, we detail three example architectures, as well as three possible models to sparsify hyperspectral datacubes.

Compressive hyperspectral imaging architectures

Single pixel hyperspectral camera

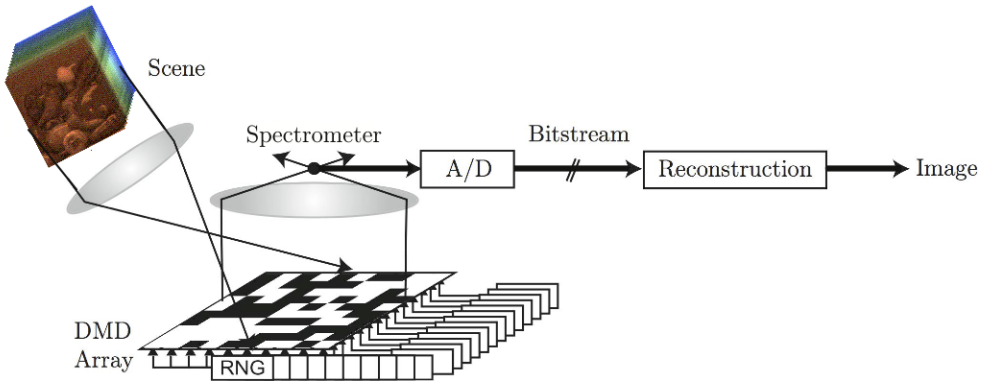
The single pixel camera uses a single photodetector to record random projections of the light emanated from the image, with the different random projections being captured in sequence. A single pixel hyperspectral camera requires a light modulating element that is reflective across the wavelengths of interest, as well as a sensor that can record the desired spectral bands separately [link]. A block diagram is shown in [link].

The single sensor consists of a single spectrometer that spans the necessary wavelength range, which replaces the photodiode. The spectrometer records the intensity of the light reflected by the modulator in each wavelength. The same digital micromirror device (DMD) provides reflectivity for wavelengths from near infrared to near ultraviolet. Thus, by converting the datacube into a vector sorted by spectral band, the matrix that operates on the data to obtain the CS measurements is represented as

Equation:

$$\Phi = \begin{bmatrix} \Phi_{x,y} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Phi_{x,y} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \Phi_{x,y} \end{bmatrix}.$$

This architecture performs multiplexing only in the spatial domain, i.e. dimensions x and y, since there is no mixing of the different spectral bands along the dimension λ.



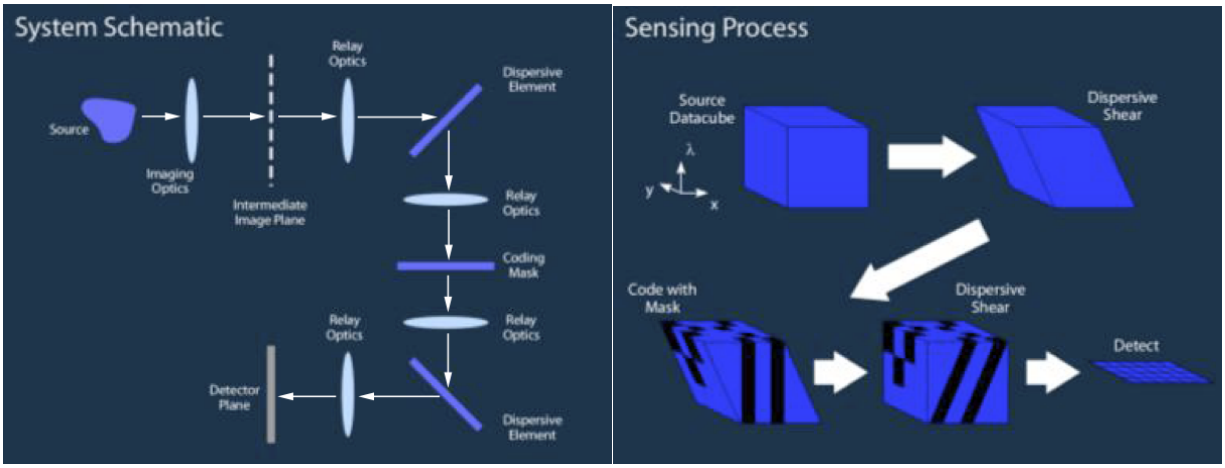
Block diagram for a single pixel hyperspectral camera. The photodiode is replaced by a spectrometer that captures the modulated light intensity for all spectral bands, for each of the CS measurements.

Dual disperser coded aperture snapshot spectral imager

The dual disperser coded aperture snapshot spectral imager (DD-CASSI), shown in [link], is an architecture that combines separate multiplexing in the spatial and spectral domain, which is then sensed by a wide-wavelength sensor/pixel array, thus flattening the spectral dimension [link].

First, a dispersive element separates the different spectral bands, which still overlap in the spatial domain. In simple terms, this element shears the datacube, with each spectral slice being displaced from the previous by a constant amount in the same spatial dimension. The resulting datacube is then masked using the coded aperture, whose effect is to "punch holes" in the sheared datacube by blocking certain pixels of light. Subsequently, a second dispersive element acts on the masked, sheared datacube; however, this element shears in the opposite direction, effectively inverting the shearing of the first dispersive element. The resulting datacube is upright, but features "sheared" holes of datacube voxels that have been masked out.

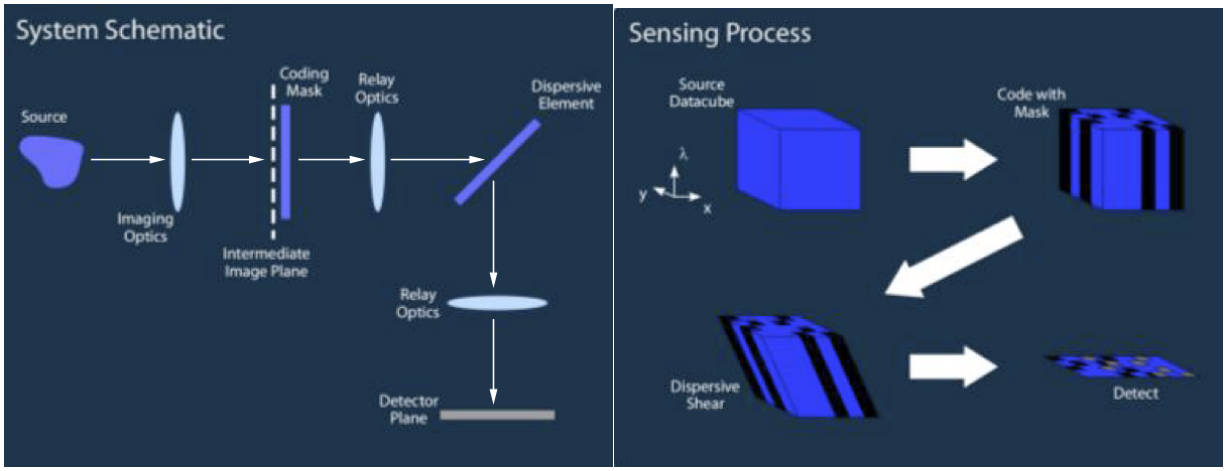
The resulting modified datacube is then received by a sensor array, which flattens the spectral dimension by measuring the sum of all the wavelengths received; the received light field resembles the target image, allowing for optical adjustments such as focusing. In this way, the measurements consist of full sampling in the spatial x and y dimensions, with an aggregation effect in the spectral λ dimension.



Dual disperser coded aperture snapshot spectral imager (DD-CASSI). (a) Schematic of the DD-CASSI components. (b) Illustration of the datacube processing performed by the components.

Single disperser coded aperture snapshot spectral imager

The single disperser coded aperture snapshot spectral imager (SD-CASSI), shown in [link], is a simplification of the DD-CASSI architecture in which the first dispersive element is removed [link]. Thus, the light field received at the sensors does not resemble the target image. Furthermore, since the shearing is not reversed, the area occupied by the sheared datacube is larger than that of the original datacube, requiring a slightly larger number of pixels for the capture.

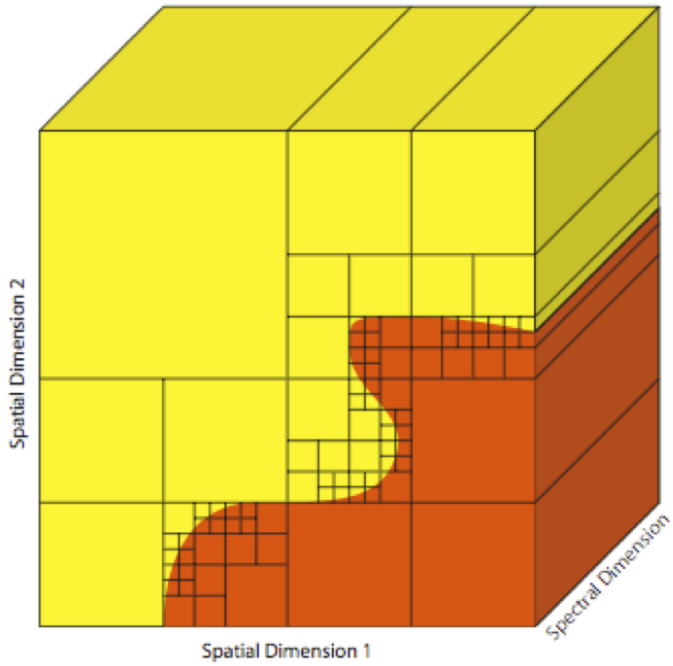


Single disperser coded aperture snapshot spectral imager (SD-CASSI). (a) Schematic of the SD-CASSI components. (b) Illustration of the datacube processing performed by the components.

Sparsity structures for hyperspectral datacubes

Dyadic Multiscale Partitioning

This sparsity structure assumes that the spectral signature for all pixels in a neighborhood is close to constant; that is, that the datacube is piecewise constant with smooth borders in the spatial dimensions. The complexity of an image is then given by the number of spatial dyadic squares with constant spectral signature necessary to accurately approximate the datacube; see [link]. A reconstruction algorithm then searches for the signal of lowest complexity (i.e., with the fewest dyadic squares) that generates compressive measurements close to those observed [link].



Example dyadic square partition for piecewise spatially constant datacube.

Spatial-only sparsity

This sparsity structure operates on each spectral band separately and assumes the same type of sparsity structure for each band [link]. The sparsity basis is drawn from those commonly used in images, such as wavelets, curvelets, or the discrete cosine basis. Since each basis operates only on a band, the resulting sparsity basis for the datacube can be represented as a block-diagonal matrix:
Equation:

$$\Psi = \begin{matrix} \Psi_{x,y} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Psi_{x,y} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \Psi_{x,y} \end{matrix} .$$

Kronecker product sparsity

This sparsity structure employs separate sparsity bases for the spatial dimensions and the spectral dimension, and builds a sparsity basis for the datacube using the Kronecker product of these two [link]:
Equation:

$$\Psi = \Psi_\lambda \otimes \Psi_{x,y} = \begin{matrix} \Psi_\lambda[1,1]\Psi_{x,y} & \Psi_\lambda[1,2]\Psi_{x,y} & \cdots \\ \Psi_\lambda[2,1]\Psi_{x,y} & \Psi_\lambda[2,2]\Psi_{x,y} & \cdots \\ \vdots & \vdots & \ddots \end{matrix} .$$

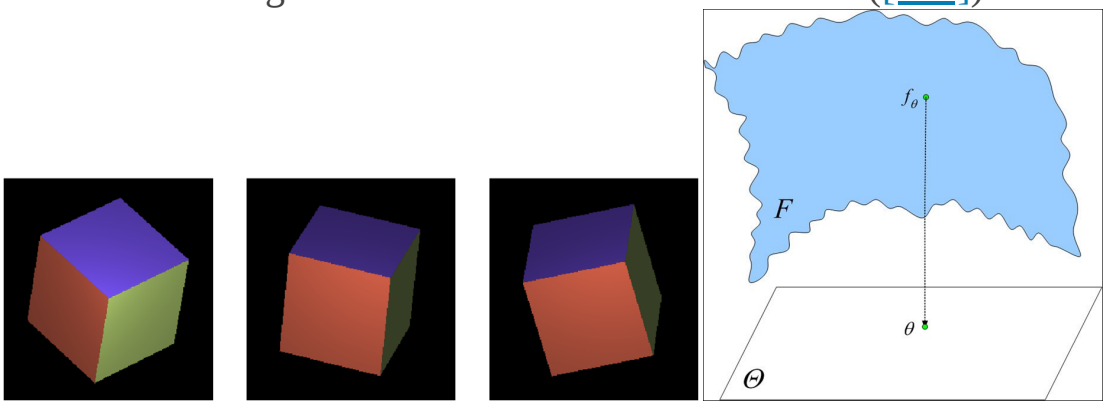
In this manner, the datacube sparsity bases simultaneously enforces both spatial and spectral structure, potentially achieving a sparsity level lower than the sums of the spatial sparsities for the separate spectral slices, depending on the level of structure between them and how well can this structure be captured through sparsity.

Summary

Compressive sensing will make the largest impact in applications with very large, high dimensional datasets that exhibit considerable amounts of structure. Hyperspectral imaging is a leading example of such applications; the sensor architectures and data structure models surveyed in this module show initial promising work in this new direction, enabling new ways of simultaneously sensing and compressing such data. For standard sensing architectures, the data structures surveyed also enable new transform coding-based compression schemes.

Compressive processing of manifold-modeled data
This module outlines the connection between compressive sensing and random projections of manifolds.

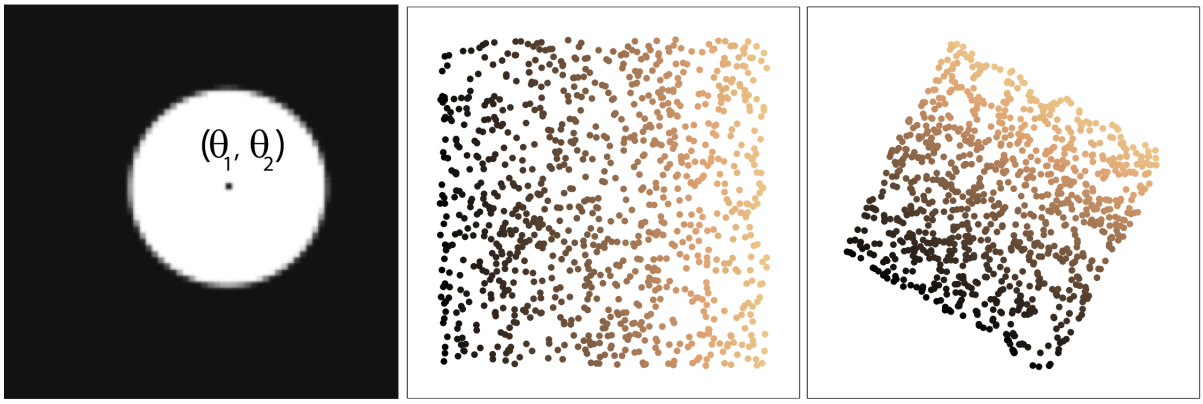
A powerful data model for many applications is the geometric notion of a low-dimensional *manifold*. Data that possesses merely K intrinsic degrees of freedom" can be assumed to lie on a K-dimensional manifold in the high-dimensional ambient space. Once the manifold model is identified, any point on it can be represented using essentially K pieces of information. For instance, suppose a stationary camera of resolution N observes a truck moving down along a straight line on a highway. Then, the set of images captured by the camera forms a 1-dimensional manifold in the image space $\mathbb{R}^N$. Another example is the set of images captured by a static camera observing a cube that rotates in 3 dimensions. ([link]).



(a) A rotating cube has 3 degrees of freedom, thus giving rise to a 3-dimensional manifold in image space. (b) Illustration of a manifold F parametrized by a $K-$dimensional vector θ.

In many applications, it is beneficial to explicitly characterize the structure (alternately, identify the parameters) of the manifold formed by a set of observed signals. This is known as *manifold learning* and has been the subject of considerable study over the last several years; well-known manifold learning algorithms include Isomap [link], LLE [link], and Hessian eigenmaps [link]. An informal example is as follows: if a 2-dimensional manifold were to be imagined as the surface of a twisted sheet

of rubber, manifold learning can be described as the process of “unraveling” the sheet and stretching it out on a 2D flat surface. [link] indicates the performance of Isomap on a simple 2-dimensional dataset comprising of images of a translating disk.

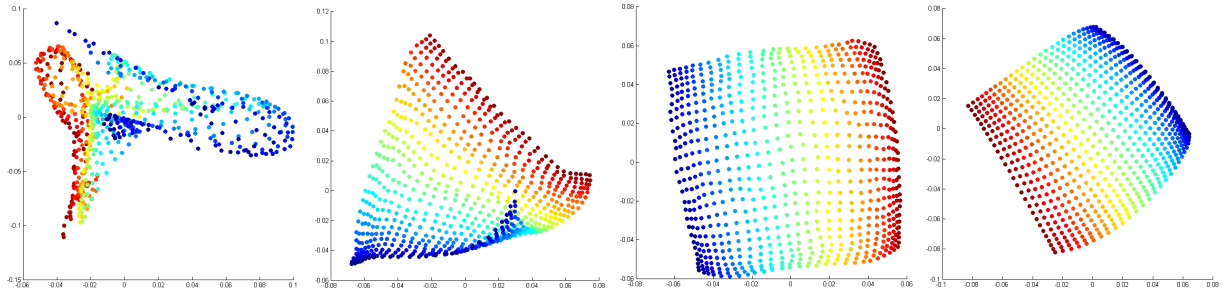


(a) Input data consisting of 1000 images of a disk shifted in $K = 2$ dimensions, parametrized by an articulation vector (θ_1, θ_2). (b) True θ_1 and θ_2 values of the sampled data. (c) Isomap embedding learned from original data in $\mathbb{R}^N$.

A *linear, nonadaptive* manifold dimensionality reduction technique has recently been introduced that employs the technique of random projections [link]. Consider a K-dimensional manifold in the ambient space $\mathbb{R}^N$ and its projection onto a random subspace of dimension $M = CK \log(N)$; note that $K < M << N$. The result of [link] is that the pairwise metric structure of sample points from is preserved with high accuracy under projection from $\mathbb{R}^N$ to $\mathbb{R}^M$. This is analogous to the result for compressive sensing of sparse signals (see "The restricted isometry property"; however, the difference is that the number of projections required to preserve the ensemble structure does *not* depend on the sparsity of the individual images, but rather on the dimension of the underlying manifold.

This result has far reaching implications; it suggests that a wide variety of signal processing tasks can be performed *directly on the random projections* acquired by these devices, thus saving valuable sensing, storage and

processing costs. In particular, this enables provably efficient manifold learning in the projected domain [link]. [link] illustrates the performance of Isomap on the translating disk dataset under varying numbers of random projections.



Isomap embeddings learned from random projections of the 625 images of shifting squares. (a) 25 random projections; (b) 50 random projections; (c) 25 random projections; (d) full data.

The advantages of random projections extend even to cases where the original data is available in the ambient space $\mathbb{R}^N$. For example, consider a wireless network of cameras observing a static scene. The set of images captured by the cameras can be visualized as living on a low-dimensional manifold in the image space. To perform joint image analysis, the following steps might be executed:

1. **Collate**: Each camera node transmits its respective captured image (of size N) to a central processing unit.
2. **Preprocess**: The central processor estimates the *intrinsic dimension* K of the underlying image manifold.
3. **Learn**: The central processor performs a nonlinear embedding of the data points – for instance, using Isomap [link] – into a K-dimensional Euclidean space, using the estimate of K from the previous step.

In situations where N is large and communication bandwidth is limited, the dominating costs will be in the first transmission/collation step. To reduce the communication expense, one may perform nonlinear image compression (such as JPEG) at each node before transmitting to the central processing. However, this requires a good deal of processing power at each

sensor, and the compression would have to be undone during the learning step, thus adding to overall computational costs.

As an alternative, every camera could encode its image by computing (either directly or indirectly) a small number of random projections to communicate to the central processor [link]. These random projections are obtained by linear operations on the data, and thus are cheaply computed. Clearly, in many situations it will be less expensive to store, transmit, and process such randomly projected versions of the sensed images. The method of random projections is thus a powerful tool for ensuring the stable embedding of low-dimensional manifolds into an intermediate space of reasonable size. It is now possible to think of settings involving a huge number of low-power devices that inexpensively capture, store, and transmit a very small number of measurements of high-dimensional data.

Inference using compressive measurements
This module provides an introduction to some simple algorithms for compressive signal processing, i.e., processing compressive measurements directly without first recovering the signal to solve an inference problem.

While the compressive sensing (CS) literature has focused almost exclusively on problems in signal reconstruction/approximation, this is frequently not necessary. For instance, in many signal processing applications (including computer vision, digital communications and radar systems), signals are acquired only for the purpose of making a detection or classification decision. Tasks such as detection do not require a reconstruction of the signal, but only require estimates of the relevant *sufficient statistics* for the problem at hand.

As a simple example, suppose a surveillance system (based on compressive imaging) observes the motion of a person across a static background. The relevant information to be extracted from the data acquired by this system would be, for example, the identity of the person, or the location of this person with respect to a predefined frame of coordinates. There are two ways of doing this:

- Reconstruct the full data using standard sparse recovery techniques and apply standard computer vision/inference algorithms on the reconstructed images.
- Develop an inference test which operates *directly* on the compressive measurements, without ever reconstructing the full images.

A crucial property that enables the design of compressive inference algorithms is the *information scalability* property of compressive measurements. This property arises from the following two observations:

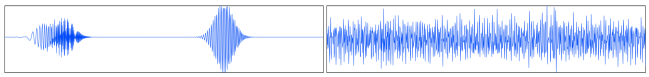
- For certain signal models, the action of a *random* linear function on the set of signals of interest preserves enough information to perform inference tasks on the observed measurements.
- The *number* of random measurements required to perform the inference task usually depends on the nature of the inference task. Informally, we observe that more sophisticated tasks require more measurements.

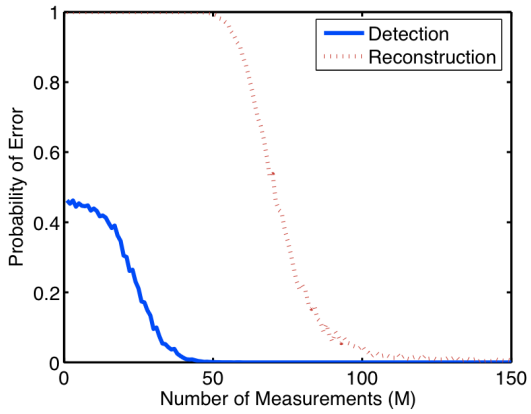
We examine three possible inference problems for which algorithms that *directly operate on the compressive measurements* can be developed: detection (determining the presence or absence of an information-bearing signal), classification (assigning the observed signal to one of two (or more) signal classes), and parameter estimation (calculating a *function* of the observed signal).

Detection

In detection one simply wishes to answer the question: is a (known) signal present in the observations? To solve this problem, it suffices to estimate a relevant *sufficient statistic*. Based on a concentration of measure inequality, it is possible to show that such sufficient statistics for a detection problem can be accurately estimated from random projections, where the quality of this estimate depends on the signal to noise ratio (SNR) [link]. We make no assumptions on the signal of interest s, and hence we can build systems capable of detecting s even when it is not known in advance. Thus, we can use random projections for dimensionality-reduction in the detection setting without knowing the relevant structure.

In the case where the class of signals of interest corresponds to a low dimensional subspace, a truncated, simplified sparse approximation can be applied as a detection algorithm; this has been dubbed as IDEA [link]. In simple terms, the algorithm will mark a detection when a large enough amount of energy from the measurements lies in the projected subspace. Since this problem does not require accurate estimation of the signal values, but rather whether it belongs in the subspace of interest or not, the number of measurements necessary is much smaller than that required for reconstruction, as shown in [link].





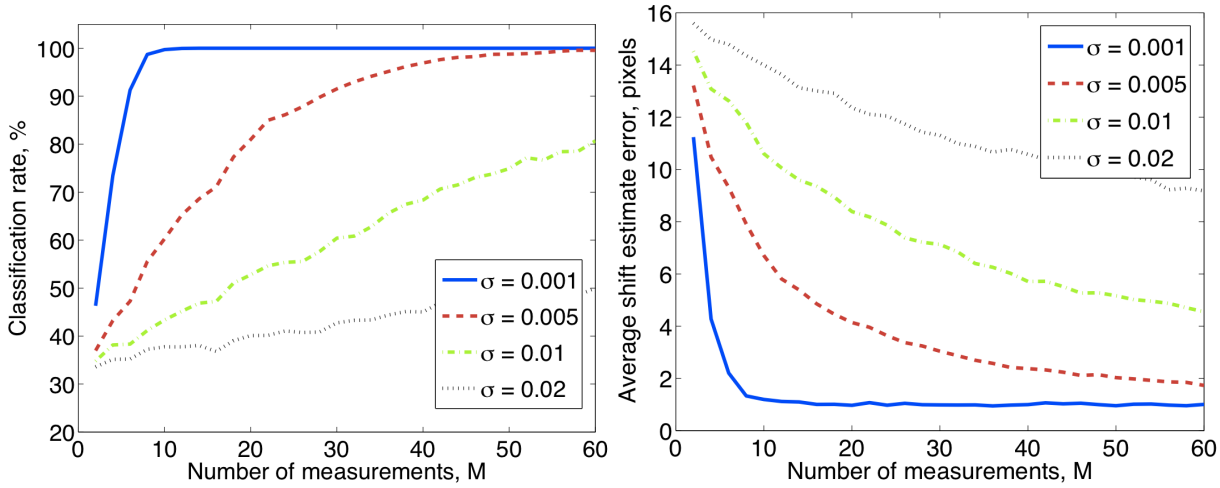
Performance for IDEA. (Top) Sample wideband chirp *signal and same chirp embedded in strong narrowband interference. (Bottom) Probability of error to reconstruct and detect chirp signals embedded in strong sinusoidal interference (*$\mathrm{SIR} = -6$ *dB) using greedy algorithms. In this case, detection requires* $3\times$ *fewer measurements and* $4\times$ *fewer computations than reconstruction for an equivalent probability of success. Taken from [link].*

Classification

Similarly, random projections have long been used for a variety of classification and clustering problems. The Johnson-Lindenstrauss Lemma is often exploited in this setting to compute approximate nearest neighbors, which is naturally related to classification. The key result that random projections result in an isometric embedding allows us to generalize this work to several new classification algorithms and settings [link].

Classification can also be performed when more elaborate models are used for the different classes. Suppose the signal/image class of interest can be modeled as a low-dimensional manifold in the ambient space. In such case it can be shown that, even under random projections, certain geometric properties of the signal class are preserved up to a small distortion; for example, interpoint Euclidean (ℓ_2) distances are preserved [link]. This enables the design of classification algorithms in the *projected* domain. One

such algorithm is known as the smashed filter [link]. As an example, under equal distribution among classes and a gaussian noise setting, the smashed filter is equivalent to building a nearest-neighbor (NN) classifier in the measurement domain. Further, it has been shown that for a $K-$dimensional manifold, $M = O(K \log N)$ measurements are sufficient to perform reliable compressive classification. Thus, the number of measurements scales as the dimension of the signal class, as opposed to the *sparsity* of the individual signal. Some example results are shown in [link](a).



Results for smashed filter image classification and parameter estimation experiments. (a) Classification rates and (b) average estimation error for varying number of measurements M and noise levels σ for a set of images of several objects under varying shifts. As M increases, the distances between the manifolds increase as well, thus increasing the noise tolerance and enabling more accurate estimation and classification. Thus, the classification and estimation performances improve as σ decreases and M increases in all cases. Taken from [link].

Estimation

Consider a signal $x \in \quad ^N$, and suppose that we wish to estimate some function $f(x)$ but only observe the measurements $y = \Phi x$, where Φ is again an $M \times N$ matrix. The data streaming community has previously analyzed this problem for many common functions, such as linear functions, ℓ_p norms, and histograms. These estimates are often based on so-called *sketches*, which can be thought of as random projections.

As an example, in the case where f is a *linear* function, one can show that the estimation error (relative to the norms of x and f) can be bounded by a constant determined by M. This result holds for a wide class of random matrices, and can be viewed as a straightforward consequence of the same concentration of measure inequality that has proven useful for CS and in proving the JL Lemma [link].

Parameter estimation can also be performed when the signal class is modeled as a low-dimensional manifold. Suppose an observed signal x can be parameterized by a $K-$dimensional parameter vector θ, where $K \ll N$. Then, it can be shown that with $0(K \log N)$ measurements, the parameter vector can be obtained via *multiscale manifold navigation* in the compressed domain [link]. Some example results are shown in [link](b).

Compressive sensor networks
This module provides an overview of applications of compressive sensing in the context of distributed sensor networks.

Sparse and compressible signals are present in many sensor network applications, such as environmental monitoring, signal field recording and vehicle surveillance. Compressive sensing (CS) has many properties that make it attractive in this settings, such as its low complexity sensing and compression, its universality and its graceful degradation. CS is robust to noise, and allows querying more nodes to obey further detail on signals as they become interesting. Packet drops also do not harm the network nearly as much as many other protocols, only providing a marginal loss for each measurement not obtained by the receiver. As the network becomes more congested, data can be scaled back smoothly.

Thus CS can enable the design of generic compressive sensors that perform random or incoherent projections.

Several methods for using CS in sensor networks have been proposed. Decentralized methods pass data throughout the network, from neighbor to neighbor, and allow the decoder to probe any subset of nodes. In contrast, centralized methods require all information to be transmitted to a centralized data center, but reduce either the amount of information that must be transmitted or the power required to do so. We briefly summarize each class below.

Decentralized algorithms

Decentralized algorithms enable the calculation of compressive measurements at each sensor in the network, thus being useful for applications where monitoring agents traverse the network during operation.

Randomized gossiping

In randomized gossiping [link], each sensor communicates M random projection of its data sample to a random set of nodes, in each stage aggregating and forwarding the observations received to a new set of random nodes. In essence, a spatial dot product is being performed as each node collects and aggregates information, compiling a sum of the weighted samples to obtain M CS measurements which becomes more accurate as more rounds of random gossiping occur. To recover the data, a basis that provides data sparsity (or at least compressibility) is required, as well as the random projections used. However, this information does not need to be known while the data is being passed.

The method can also be applied when each sensor observes a compressible signal. In this case, each sensor computes multiple random projections of the data and transmits them using randomized gossiping to the rest of the network. A potential drawback of this technique is the amount of storage required per sensor, as it could be considerable for large networks . In this case, each sensor can store the data from only a subset of the sensors, where each group of sensors of a certain size will be known to contain CS measurements for all the data in the network. To maintain a constant error as the network size grows, the number of transmissions becomes $\Theta\left(kMn^2\right)$, where k represents the number of groups in which the data is partitioned, M is the number of values desired from each sensor and n are the number of nodes in the network. The results can be improved by using geographic gossiping algorithms [link].

Distributed sparse random projections

A second method modifies the randomized gossiping approach by limiting the number of communications each node must perform, in order to reduce overall power consumption [link]. Each data node takes M projections of its data, passing along information to a small set of L neighbors, and summing the observations; the resulting CS measurements are sparse, since $N - L$ of each row's entries will be zero. Nonetheless, these projections can still be used as CS measurements with quality similar to that of full random projections. Since the CS measurement matrix formed by the data nodes is sparse, a relatively small amount of communication is performed

by each encoding node and the overall power required for transmission is reduced.

Centralized algorithms

Decentralized algorithms are used when the sensed data must be routed to a single location; this architecture is common in sensor networks were low power, simple nodes perform sensing and a powerful central location performs data processing.

Compressive wireless sensing

Compressive wireless sensing (CWS) emphasizes the use of synchronous communication to reduce the transmission power of each sensor [link]. In CWS, each sensor calculates a noisy projection of their data sample. Each sensor then transmits the calculated value by analog modulation and transmission of a communication waveform. The projections are aggregated at the central location by the receiving antenna, with further noise being added. In this way, the fusion center receives the CS measurements, from which it can perform reconstruction using knowledge of the random projections.

A drawback of this method is the required accurate synchronization. Although CWS is constraining the power of each node, it is also relying on constructive interference to increase the power received by the data center. The nodes themselves must be accurately synchronized to know when to transmit their data. In addition, CWS assumes that the nodes are all at approximately equal distances from the fusion center, an assumption that is acceptable only when the receiver is far away from the sensor network. Mobile nodes could also increase the complexity of the transmission protocols. Interference or path issues also would have a large effect on CWS, limiting its applicability.

If these limitations are addressed for a suitable application, CWS does offer great power benefits when very little is known about the data beyond

sparsity in a fixed basis. Distortion will be proportional to $M^{-2\alpha/(2\alpha+1)}$, where α is some positive constant based on the network structure. With much more a priori information about the sensed data, other methods will achieve distortions proportional to $M^{-2\alpha}$.

Distributed compressive sensing

Distributed Compressive Sensing (DCS) provides several models for combining neighboring sparse signals, relying on the fact that such sparse signals may be similar to each other, a concept that is termed joint sparsity [link]. In an example model, each signal has a common component and a local innovation, with the commonality only needing to be encoded once while each innovation can be encoded at a lower measurement rate. Three different joint sparsity models (JSMs) have been developed:

1. Both common signal and innovations are sparse;
2. Sparse innovations with shared sparsity structure;
3. Sparse innovations and dense common signal.

Although JSM 1 would seem preferable due to the relatively limited amount of data, only JSM 2 is computationally feasible for large sensor networks; it has been used in many applications [link]. JSMs 1 and 3 can be solved using a linear program, which has cubic complexity on the number of sensors in the network.

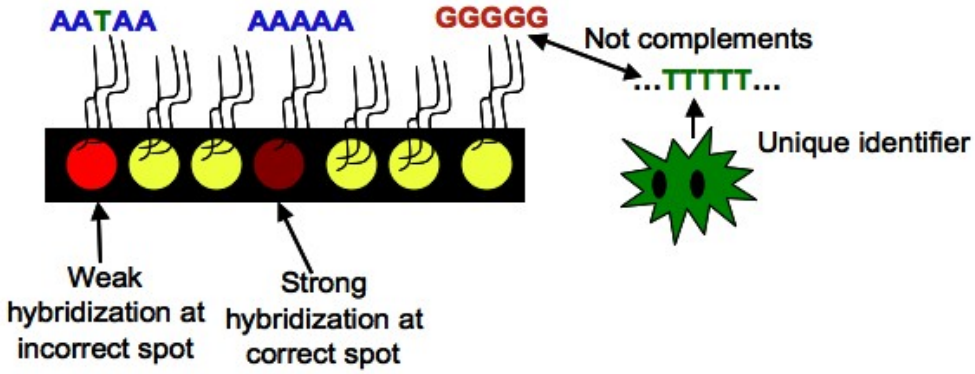
DCS, however, does not address the communication or networking necessary to transmit the measurements to a central location; it relies on standard communication and networking techniques for measurement transmission, which can be tailored to the specific network topology.

Genomic sensing

This module describes the application of compressive sensing to the design of new kinds of DNA microarray probes.

Biosensing of pathogens is a research area of high consequence. An accurate and rapid biosensing paradigm has the potential to impact several fields, including healthcare, defense and environmental monitoring. In this module we address the concept of biosensing based on compressive sensing (CS) via the *Compressive Sensing Microarray* (CSM), a DNA microarray adapted to take CS-style measurements.

DNA microarrays are a frequently applied solution for microbe sensing; they have a significant edge over competitors due to their ability to sense many organisms in parallel [link], [link]. A DNA microarray consists of genetic sensors or *spots*, each containing DNA sequences termed *probes*. From the perspective of a microarray, each DNA sequence can be viewed as a sequence of four DNA bases $\{A, T, G, C\}$ that tend to bind with one another in complementary pairs: A with T and G with C. Therefore, a DNA subsequence in a target organism's genetic sample will tend to bind or "hybridize" with its complementary subsequence on a microarray to form a stable structure. The target DNA sample to be identified is fluorescently tagged before it is flushed over the microarray. The extraneous DNA is washed away so that only the bound DNA is left on the array. The array is then scanned using laser light of a wavelength designed to trigger fluorescence in the spots where binding has occurred. A specific pattern of array spots will fluoresce, which is then used to infer the genetic makeup in the test sample.



Cartoon of traditional DNA microarray showing strong

and weak hybridization of the unique pathogen identifier
at different microarray spots

There are three issues with the traditional microarray design. Each spot consists of probes that can uniquely identify only one target of interest (each spot contains multiple copies of a probe for robustness.) The first concern with this design is that very often the targets in a test sample have similar base sequences, causing them to hybridize with the wrong probe (see [link]). These cross-hybridization events lead to errors in the array readout. Current microarray design methods do not address cross-matches between similar DNA sequences.

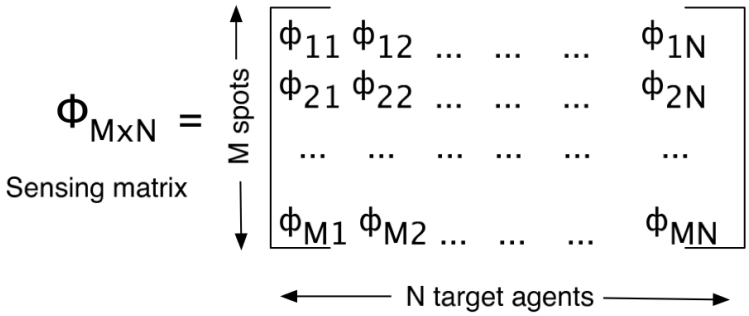
The second concern in choosing unique identifier based DNA probes is its restriction on the number of organisms that can be identified. In typical biosensing applications multiple organisms must be identified; therefore a large number of DNA targets requires a microarray with a large number of spots. In fact, there are over 1000 known harmful microbes, many with more than 100 strains. The implementation cost and processing speed of microarray data is directly related to its number of spots, representing a significant problem for commercial deployment of microarray-based biosensors. As a consequence readout systems for traditional DNA arrays cannot be miniaturized or implemented using electronic components and require complicated fluorescent tagging.

The third concern is the inefficient utilization of the large number of array spots in traditional microarrays. Although the number of potential agents in a sample is very large, *not all agents* are expected to be present in a significant concentration at a given time and location, or in an air/water/soil sample to be tested. Therefore, in a traditionally designed microarray only a small fraction of spots will be active at a given time, corresponding to the few targets present.

To combat these problems, a Compressive Sensing DNA Microarray (CSM) uses “combinatorial testing sensors” in order to reduce the number of sensor spots [link], [link], [link]. Each spot in the CSM identifies a *group* of target

organisms, and several spots together generate a unique pattern identifier for a single target. (See also "Group testing and data stream algorithms".) Designing the probes that perform this combinatorial sensing is the essence of the microarray design process, and what we aim to describe in this module.

To obtain a CS-type measurement scheme, we can choose each probe in a CSM to be a group identifier such that the readout of each probe is a probabilistic combination of all the targets in its group. The probabilities are representative of each probe's hybridization affinity (or stickiness) to those targets in its group; the targets that are not in its group have low affinity to the probe. The readout signal at each spot of the microarray is a linear combination of hybridization affinities between its probe sequence and each of the target agents.



Structure of the CSM sensing matrix Φ with M spots identifying N targets

[link] illustrates the sensing process. To formalize, we assume there are M spots on the CSM and N targets; we have far fewer spots than target agents. For $1 \leq i \leq M$ and $1 \leq j \leq N$, the probe at spot i hybridizes with target j with affinity $\varphi_{i,j}$. The target j occurs in the tested DNA sample with concentration x_j, so that the total hybridization of spot i is $y_i = \sum_{j=1}^{N} \varphi_{i,j} x_j = \varphi_i x$, where φ_j and x are a row and column vector, respectively. The resulting measured microarray signal intensity vector $y = \{y_i\}_i = 1, ..., M$ fits the CS measurement model $y = \Phi x$.

While group testing has previously been proposed for microarrays [link], the sparsity in the target signal is key in applying CS. The chief advantage of a CS-based approach over regular group testing is in its information scalability. We are able to not just detect, but *estimate* the target signal with a reduced number of measurements similar to that of group testing [link]. This is important since there are always minute quantities of certain pathogens in the environment, but it is only their large concentrations that may be harmful to us. Furthermore, we are able to use CS recovery methods such as Belief Propagation that decode x while accounting for experimental noise and measurement nonlinearities due to excessive target molecules [link].

Sub-Gaussian random variables

In this module we introduce the sub-Gaussian and strictly sub-Gaussian distributions. We provide some simple examples and illustrate some of the key properties of sub-Gaussian random variables.

A number of distributions, notably Gaussian and Bernoulli, are known to satisfy certain concentration of measure inequalities. We will analyze this phenomenon from a more general perspective by considering the class of sub-Gaussian distributions [link].

> A random variable X is called *sub-Gaussian* if there exists a constant $c > 0$ such that
> **Equation:**
>
> $$\mathbb{E}(\exp\,(Xt)) \leq \exp\,\left(c^2 t^2/2\right)$$
>
> holds for all $t \in \mathbb{R}$. We use the notation $X \sim \mathrm{Sub}\left(c^2\right)$ to denote that X satisfies [link].

The function $\mathbb{E}(\exp\,(Xt))$ is the *moment-generating function* of X, while the upper bound in [link] is the moment-generating function of a Gaussian random variable. Thus, a sub-Gaussian distribution is one whose moment-generating function is bounded by that of a Gaussian. There are a tremendous number of sub-Gaussian distributions, but there are two particularly important examples:

Example:
If $X \sim \mathcal{N}\left(0, \sigma^2\right)$, i.e., X is a zero-mean Gaussian random variable with variance σ^2, then $X \sim \mathrm{Sub}\left(\sigma^2\right)$. Indeed, as mentioned above, the moment-generating function of a Gaussian is given by $\mathbb{E}(\exp\,(Xt)) = \exp\,\left(\sigma^2 t^2/2\right)$, and thus [link] is trivially satisfied.

Example:
If X is a zero-mean, bounded random variable, i.e., one for which there exists a constant B such that $|X| \leq B$ with probability 1, then $X \sim \mathrm{Sub}\left(B^2\right)$.

A common way to characterize sub-Gaussian random variables is through analyzing their moments. We consider only the mean and variance in the following elementary lemma, proven in [link].
(Buldygin-Kozachenko [link])

If $X \sim \mathrm{Sub}\left(c^2\right)$ then,
Equation:

$$\mathbb{E}(X) = 0$$

and
Equation:

$$\mathbb{E}\left(X^2\right) \leq c^2.$$

[link] shows that if $X \sim \mathrm{Sub}\left(c^2\right)$ then $\mathbb{E}\left(X^2\right) \leq c^2$. In some settings it will be useful to consider a more restrictive class of random variables for which this inequality becomes an equality.

> A random variable X is called *strictly sub-Gaussian* if $X \sim \mathrm{Sub}\left(\sigma^2\right)$ where $\sigma^2 = \mathbb{E}\left(X^2\right)$, i.e., the inequality
> **Equation:**
>
> $$\mathbb{E}(\exp\left(Xt\right)) \leq \exp\left(\sigma^2 t^2/2\right)$$
>
> holds for all $t \in \mathbb{R}$. To denote that X is strictly sub-Gaussian with variance σ^2, we will use the notation $X \sim \mathrm{SSub}\left(\sigma^2\right)$.

Example:
If $X \sim \mathcal{N}\left(0, \sigma^2\right)$, then $X \sim \operatorname{SSub}\left(\sigma^2\right)$.

Example:
If $X \sim \mathrm{U}(-1,1)$, i.e., X is uniformly distributed on the interval $[-1,1]$, then $X \sim \operatorname{SSub}(1/3)$.

Example:
Now consider the random variable with distribution such that
Equation:

$$\mathbb{P}(X=1)=\mathbb{P}(X=-1)=\frac{1-s}{2}, \quad \mathbb{P}(X=0)=s, \quad s \in [0,1).$$

For any $s \in [0,2/3]$, $X \sim \operatorname{SSub}(1-s)$. For $s \in (2/3,1)$, X is not strictly sub-Gaussian.

We now provide an equivalent characterization for sub-Gaussian and strictly sub-Gaussian random variables, proven in [link], that illustrates their concentration of measure behavior.
(Buldygin-Kozachenko [link])

A random variable $X \sim \operatorname{Sub}\left(c^2\right)$ if and only if there exists a $t_0 \geq 0$ and a constant $a \geq 0$ such that
Equation:

$$\mathbb{P}(|X| \geq t) \leq 2 \exp \quad -\frac{t^2}{2a^2}$$

for all $t \geq t_0$. Moreover, if $X \sim \mathrm{SSub}\left(\sigma^2\right)$, then [link] holds for all $t > 0$ with $a = \sigma$.

Finally, sub-Gaussian distributions also satisfy one of the fundamental properties of a Gaussian distribution: the sum of two sub-Gaussian random variables is itself a sub-Gaussian random variable. This result is established in more generality in the following lemma.

Suppose that $X = [X_1, X_2, ..., X_N]$, where each X_i is independent and identically distributed (i.i.d.) with $X_i \sim \mathrm{Sub}\left(c^2\right)$. Then for any $\alpha \in \mathbb{R}^N$, $\langle X, \alpha \rangle \sim \mathrm{Sub}\ \ c^2 \| \alpha \|_2^2$. Similarly, if each $X_i \sim \mathrm{SSub}\left(\sigma^2\right)$, then for any $\alpha \in \mathbb{R}^N$, $\langle X, \alpha \rangle \sim \mathrm{SSub}\ \ \sigma^2 \| \alpha \|_2^2$.

Since the X_i are i.i.d., the joint distribution factors and simplifies as:
Equation:

$$\begin{aligned}
\mathbb{E}\ \ \exp\ \ t \sum_{i=1}^{N} \alpha_i X_i \quad &= \mathbb{E}\ \ \prod_{i=1}^{N} \exp\left(t \alpha_i X_i\right) \\
&= \prod_{i=1}^{N} \mathbb{E}(\exp\left(t \alpha_i X_i\right)) \\
&\leq \prod_{i=1}^{N} \exp\ \ c^2 (\alpha_i t)^2 / 2 \\
&= \exp \sum_{i=1}^{N} \alpha_i^2\ \ c^2 t^2 / 2\ \ .
\end{aligned}$$

If the X_i are strictly sub-Gaussian, then the result follows by setting $c^2 = \sigma^2$ and observing that $\mathbb{E}\ \ \langle X, \alpha \rangle^2\ \ = \sigma^2 \| \alpha \|_2^2$.

Concentration of measure for sub-Gaussian random variables
This module establishes concentration bounds for sub-Gaussian vectors and matrices.

Sub-Gaussian distributions have a close relationship to the concentration of measure phenomenon [link]. To illustrate this, we note that we can combine Lemma 2 and Theorem 1 from "Sub-Gaussian random variables" to obtain deviation bounds for weighted sums of sub-Gaussian random variables. For our purposes, however, it will be more enlightening to study the *norm* of a vector of sub-Gaussian random variables. In particular, if X is a vector where each X_i is i.i.d. with $X_i \sim \text{Sub}\,(c)$, then we would like to know how $\|\ X\ \|_2$ deviates from its mean.

In order to establish the result, we will make use of Markov's inequality for nonnegative random variables.
(Markov's Inequality)

For any nonnegative random variable X and $t > 0$,
Equation:

$$\mathbb{P}\left(X \geq t\right) \leq \frac{\mathbb{E}(X)}{t}.$$

Let $f(x)$ denote the probability density function for X.
Equation:

$$\mathbb{E}\left(X\right) = \int_0^\infty x f\left(x\right) dx \geq \int_t^\infty x f\left(x\right) dx \geq \int_t^\infty t f\left(x\right) dx = t\mathbb{P}\left(X \geq t\right).$$

In addition, we will require the following bound on the exponential moment of a sub-Gaussian random variable.

Suppose $X \sim \text{Sub}\left(c^2\right)$. Then
Equation:

$$\mathbb{E}\left(\exp\ \left(\lambda X^2/2c^2\right)\right) \leq \frac{1}{\sqrt{1-\lambda}},$$

for any $\lambda \in [0, 1)$.

First, observe that if $\lambda = 0$, then the lemma holds trivially. Thus, suppose that $\lambda \in (0, 1)$. Let $f(x)$ denote the probability density function for X. Since X is sub-Gaussian, we have by definition that
Equation:

$$\int_{-\infty}^{\infty} \exp\left(tx\right) f\left(x\right) dx \leq \exp\left(c^2 t^2 / 2\right)$$

for any $t \in \mathbb{R}$. If we multiply by $\exp\left(-c^2 t^2 / 2\lambda\right)$, then we obtain
Equation:

$$\int_{-\infty}^{\infty} \exp\left(tx - c^2 t^2 / 2\lambda\right) f\left(x\right) dx \leq \exp\left(c^2 t^2 \left(\lambda - 1\right) / 2\lambda\right).$$

Now, integrating both sides with respect to t, we obtain
Equation:

$$\int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \exp\left(tx - c^2 t^2 / 2\lambda\right) dt \right) f\left(x\right) dx \leq \int_{-\infty}^{\infty} \exp\left(c^2 t^2 \left(\lambda - 1\right) / 2\lambda\right) dt,$$

which reduces to
Equation:

$$\frac{1}{c} \sqrt{2\pi\lambda} \int_{-\infty}^{\infty} \exp\left(\lambda x^2 / 2c^2\right) f\left(x\right) dx \leq \frac{1}{c} \sqrt{\frac{2\pi\lambda}{1 - \lambda}}.$$

This simplifies to prove the lemma.

We now state our main theorem, which generalizes the results of [link] and uses substantially the same proof technique.

Suppose that $X = [X_1, X_2, ..., X_M]$, where each X_i is i.i.d. with $X_i \sim \text{Sub}\left(c^2\right)$ and $\mathbb{E}\left(X_i^2\right) = \sigma^2$. Then
Equation:

$$\mathbb{E}\Big(\parallel X \parallel_2^2\Big) = M\sigma^2.$$

Moreover, for any $\alpha \in (0,1)$ and for any $\beta \in \left[c^2/\sigma^2, \beta_{\max}\right]$, there exists a constant $\kappa^* \geq 4$ depending only on $\beta_{\max}$ and the ratio σ^2/c^2 such that
Equation:

$$\mathbb{P}\Big(\parallel X \parallel_2^2 \leq \alpha M\sigma^2\Big) \leq \exp\left(-M(1-\alpha)^2/\kappa^*\right)$$

and
Equation:

$$\mathbb{P}\Big(\parallel X \parallel_2^2 \geq \beta M\sigma^2\Big) \leq \exp\left(-M(\beta-1)^2/\kappa^*\right).$$

Since the X_i are independent, we obtain
Equation:

$$\mathbb{E}\Big(\parallel X \parallel_2^2\Big) = \sum_{i=1}^{M} \mathbb{E}\big(X_i^2\big) = \sum_{i=1}^{M} \sigma^2 = M\sigma^2$$

and hence [link] holds. We now turn to [link] and [link]. Let us first consider [link]. We begin by applying Markov's inequality:
Equation:

$$\begin{aligned}
\mathbb{P}\Big(\parallel X \parallel_2^2 \geq \beta M\sigma^2\Big) &= \mathbb{P}\Big(\exp\left(\lambda \parallel X \parallel_2^2\right) \geq \exp\left(\lambda\beta M\sigma^2\right)\Big) \\
&\leq \frac{\mathbb{E}\big(\exp\left(\lambda \parallel X \parallel_2^2\right)\big)}{\exp\left(\lambda\beta M\sigma^2\right)} \\
&= \frac{\prod_{i=1}^{M}\mathbb{E}\big(\exp\left(\lambda X_i^2\right)\big)}{\exp\left(\lambda\beta M\sigma^2\right)}.
\end{aligned}$$

Since $X_i \sim \mathrm{Sub}\left(c^2\right)$, we have from [link] that
Equation:

$$\mathbb{E}\big(\exp\left(\lambda X_i^2\right)\big) = \mathbb{E}\big(\exp\left(2c^2\lambda X_i^2/2c^2\right)\big) \leq \frac{1}{\sqrt{1-2c^2\lambda}}.$$

Thus,
Equation:

$$\prod_{i=1}^{M}\mathbb{E}\big(\exp\left(\lambda X_i^2\right)\big) \leq \left(\frac{1}{1-2c^2\lambda}\right)^{M/2}$$

and hence
Equation:

$$\mathbb{P}\Big(\parallel X \parallel_2^2 \geq \beta M\sigma^2\Big) \leq \left(\frac{\exp\left(-2\lambda\beta\sigma^2\right)}{1-2c^2\lambda}\right)^{M/2}.$$

By setting the derivative to zero and solving for λ, one can show that the optimal λ is
Equation:

$$\lambda = \frac{\beta\sigma^2 - c^2}{2c^2\sigma^2\left(1+\beta\right)}.$$

Plugging this in we obtain
Equation:

$$\mathbb{P}\Big(\parallel X \parallel_2^2 \geq \beta M\sigma^2\Big) \leq \left(\beta\frac{\sigma^2}{c^2}\,\exp\left(1-\beta\frac{\sigma^2}{c^2}\right)\right)^{M/2}.$$

Similarly,
Equation:

$$\mathbb{P}\Big(\parallel X \parallel_2^2 \leq \alpha M\sigma^2\Big) \leq \left(\alpha\frac{\sigma^2}{c^2}\,\exp\left(1-\alpha\frac{\sigma^2}{c^2}\right)\right)^{M/2}.$$

In order to combine and simplify these inequalities, note that if we define
Equation:

$$\kappa^{*} =\max \left(4, 2 \frac{\left(\beta_{\max} \sigma^{2} / c-1\right)^{2}}{\left(\beta_{\max} \sigma^{2} / c-1\right)-\log \left(\beta_{\max} \sigma^{2} / c\right)}\right)$$

then we have that for any $\gamma \in \left[0, \beta_{\max} \sigma^{2} / c\right]$ we have the bound
Equation:

$$\log (\gamma) \leq(\gamma-1)-\frac{2(\gamma-1)^{2}}{\kappa^{*}},$$

and hence
Equation:

$$\gamma \leq \exp \left((\gamma-1)-\frac{2(\gamma-1)^{2}}{\kappa^{*}}\right).$$

By setting $\gamma = \alpha\sigma^2/c^2$, [link] reduces to yield [link]. Similarly, setting $\gamma = \beta\sigma^2/c^2$ establishes [link].

This result tells us that given a vector with entries drawn according to a sub-Gaussian distribution, we can expect the norm of the vector to concentrate around its expected value of $M\sigma^2$ with exponentially high probability as M grows. Note, however, that the range of allowable choices for β in [link] is limited to $\beta \geq c^2/\sigma^2 \geq 1$. Thus, for a general sub-Gaussian distribution, we may be unable to achieve an arbitrarily tight concentration. However, recall that for strictly sub-Gaussian distributions we have that $c^2 = \sigma^2$, in which there is no such restriction. Moreover, for strictly sub-Gaussian distributions we also have the following useful corollary.[footnote]
[link] exploits the strictness in the strictly sub-Gaussian distribution twice — first to ensure that $\beta \in (1, 2]$ is an admissible range for β and then to simplify the computation of κ^{*}. One could easily establish a different version of this corollary for non-strictly sub-Gaussian vectors but for which we consider a more restricted range of ϵ provided that $c^2/\sigma^2 < 2$. However, since most of the distributions of interest in this thesis are indeed strictly sub-Gaussian, we do not pursue this route.

Note also that if one is interested in very small ϵ, then there is considerable room for improvement in the constant C^*.

Suppose that $X = [X_1, X_2, ..., X_M]$, where each X_i is i.i.d. with $X_i \sim \text{SSub}\left(\sigma^2\right)$. Then
Equation:

$$\mathbb{E}\left(\| X \|_2^2\right) = M\sigma^2$$

and for any $\epsilon > 0$,
Equation:

$$\mathbb{P}\left(\left|\| X \|_2^2 - M\sigma^2\right| \geq \epsilon M\sigma^2\right) \leq 2 \exp\left(-\frac{M\epsilon^2}{\kappa^*}\right)$$

with $\kappa^* = 2/\left(1 - \log\left(2\right)\right) \approx 6.52$.

Since each $X_i \sim \text{SSub}\left(\sigma^2\right)$, we have that $X_i \sim \text{Sub}\left(\sigma^2\right)$ and $\mathbb{E}\left(X_i^2\right) = \sigma^2$, in which case we may apply [link] with $\alpha = 1 - \epsilon$ and $\beta = 1 + \epsilon$. This allows us to simplify and combine the bounds in [link] and [link] to obtain [link]. The value of κ^* follows from the observation that $1 + \epsilon \leq 2$ so that we can set $\beta_{\max} = 2$.

Finally, from [link] we also have the following additional useful corollary. This result generalizes the main results of [link] to the broader family of general strictly sub-Gaussian distributions via a much simpler proof.

Suppose that Φ is an $M \times N$ matrix whose entries φ_{ij} are i.i.d. with $\varphi_{ij} \sim \text{SSub}\left(1/M\right)$. Let $Y = \Phi x$ for $x \in \mathbb{R}^N$. Then for any $\epsilon > 0$, and any $x \in \mathbb{R}^N$,
Equation:

$$\mathbb{E}\left(\| Y \|_2^2\right) = \| x \|_2^2$$

and
Equation:

$$\mathbb{P}\left(\left|\| Y \|_2^2 - \| x \|_2^2\right| \geq \epsilon \| x \|_2^2\right) \leq 2 \exp\left(-\frac{M\epsilon^2}{\kappa^*}\right)$$

with $\kappa^* = 2/\left(1 - \log\left(2\right)\right) \approx 6.52$.

Let φ_i denote the i^{th} row of Φ. Observe that if Y_i denotes the first element of Y, then $Y_i = \langle \varphi_i, x \rangle$, and thus by Lemma 2 from "Sub-Gaussian random variables", $Y_i \sim \text{SSub}\left(\| x \|_2^2 / M\right)$. Applying [link] to the M-dimensional random vector Y, we obtain [link].

Proof of the RIP for sub-Gaussian matrices
In this module we provide a proof that sub-Gaussian matrices satisfy the restricted isometry property.

We now show how to exploit the concentration of measure properties of sub-Gaussian distributions to provide a simple proof that sub-Gaussian matrices satisfy the restricted isometry property (RIP). Specifically, we wish to show that by constructing an $M \times N$ matrix Φ at random with M sufficiently large, then with high probability there exists a $\delta_K \in (0,1)$ such that
Equation:

$$(1-\delta_K)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\delta_K)\|x\|_2^2$$

holds for all $x \in \Sigma_K$ (where Σ_K denotes the set of all signals x with at most K nonzeros).

We begin by observing that if all we require is that $\delta_{2K} > 0$, then we may set $M = 2K$ and draw a Φ according to a Gaussian distribution, or indeed any continuous univariate distribution. In this case, with probability 1, any subset of $2K$ columns will be linearly independent, and hence all subsets of $2K$ columns will be bounded below by $1 - \delta_{2K}$ where $\delta_{2K} > 0$. However, suppose we wish to know the constant δ_{2K}. In order to find the value of the constant we must consider all possible $\binom{N}{K}$ K-dimensional subspaces of $\mathbb{R}^N$. From a computational perspective, this is impossible for any realistic values of N and K. Moreover, in light of the lower bounds we described earlier in this course, the actual value of δ_{2K} in this case is likely to be very close to 1. Thus, we focus instead on the problem of achieving the RIP of order $2K$ for a specified constant δ_{2K}.

To ensure that the matrix will satisfy the RIP, we will impose two conditions on the random distribution. First, we require that the distribution is sub-Gaussian. In order to simplify our argument, we will use the simpler results stated in Corollary 2 from "Concentration of measure for sub-Gaussian random variables", which we briefly recall.

Suppose that Φ is an $M \times N$ matrix whose entries φ_{ij} are i.i.d. with $\varphi_{ij} \sim \mathrm{SSub}\,(1/M)$. Let $Y = \Phi x$ for $x \in \mathbb{R}^N$. Then for any $\epsilon > 0$, and any $x \in \mathbb{R}^N$,
Equation:

$$\mathbb{E}\left(\| Y \|_2^2\right) = \| x \|_2^2$$

and
Equation:

$$\mathbb{P}\left(\left|\| Y \|_2^2 - \| x \|_2^2\right| \geq \epsilon \| x \|_2^2\right) \leq 2 \exp\left(-\frac{M\epsilon^2}{\kappa^*}\right)$$

with $\kappa^* = 2/\left(1 - \log\left(2\right)\right) \approx 6.52$.

By exploiting this theorem, we assume that the distribution used to construct Φ is *strictly* sub-Gaussian. This is done simply to yield more concrete constants. The argument could easily be modified to establish a similar result for general sub-Gaussian distributions by instead using Theorem 2 from "Concentration of measure for sub-Gaussian random variables".

Our second condition is that we require that the distribution yield a matrix that is approximately norm-preserving, which will require that
Equation:

$$\mathbb{E}\left(\varphi_{ij}^2\right) = \frac{1}{M},$$

and hence the variance is $1/M$.

We shall now show how the concentration of measure inequality in [link] can be used together with covering arguments to prove the RIP for sub-Gaussian random matrices. Our general approach will be to construct nets of points in each K-dimensional subspace, apply [link] to all of these points through a union bound, and then extend the result from our finite set of points to all possible K-dimensional signals. Thus, in order to prove the result, we will require the following upper bound on the number of points required to construct the nets of points. (For an overview of results similar to [link] and of various related concentration of measure results, we refer the reader to the excellent introduction of [link].)

Let $\epsilon \in (0, 1)$ be given. There exists a set of points Q such that $\| q \|_2 = 1$ for all $q \in Q$, $|Q| \leq (3/\epsilon)^K$, and for any $x \in \mathbb{R}^K$ with $\| x \|_2 = 1$ there is a point $q \in Q$ satisfying $\| x - q \|_2 \leq \epsilon$.

We construct Q in a greedy fashion. We first select an arbitrary point $q_1 \in \mathbb{R}^K$ with $\| q_1 \|_2 = 1$. We then continue adding points to Q so that at step i we add a point $q_i \in \mathbb{R}^K$ with $\| q_i \|_2 = 1$ which satisfies $\| q_i - q_j \|_2 > \epsilon$ for all $j < i$. This continues until we can add no more points (and hence for any $x \in \mathbb{R}^K$ with $\| x \|_2 = 1$ there is a point $q \in Q$ satisfying $\| x - q \|_2 \leq \epsilon$.) Now we wish to bound $|Q|$. Observe that if we center balls of radius $\epsilon/2$ at each point in Q, then these balls are disjoint and lie within a ball of radius $1 + \epsilon/2$. Thus, if $B^K(r)$ denotes a ball of radius r in $\mathbb{R}^K$, then
Equation:

$$|Q| \cdot \mathrm{Vol}\big(B^K(\epsilon/2)\big) \leq \mathrm{Vol}\big(B^K(1+\epsilon/2)\big)$$

and hence
Equation:

$$\begin{aligned} |Q| &\leq \frac{\mathrm{Vol}\big(B^K(1+\epsilon/2)\big)}{\mathrm{Vol}(B^K(\epsilon/2))} \\ &= \frac{(1+\epsilon/2)^K}{(\epsilon/2)^K} \\ &\leq (3/\epsilon)^K. \end{aligned}$$

We now turn to our main theorem, which is based on the proof given in [link].

Fix $\delta \in (0,1)$. Let Φ be an $M \times N$ random matrix whose entries φ_{ij} are i.i.d. with $\varphi_{ij} \sim \mathrm{SSub}(1/M)$. If
Equation:

$$M \geq \kappa_1 K \log\left(\frac{N}{K}\right),$$

then Φ satisfies the RIP of order K with the prescribed δ with probability exceeding $1 - 2e^{-\kappa_2 M}$, where $\kappa_1 > 1$ is arbitrary and $\kappa_2 = \delta^2/2\kappa^* - 1/\kappa_1 - \log(42e/\delta)$.

First note that it is enough to prove [link] in the case $\| x \|_2 = 1$, since Φ is linear. Next, fix an index set $T \subset \{1, 2, ..., N\}$ with $|T| = K$, and let X_T denote the K-dimensional subspace spanned by the columns of Φ_T. We choose a finite set of

points Q_T such that $Q_T \subseteq X_T$, $\| q \|_2 = 1$ for all $q \in Q_T$, and for all $x \in X_T$ with $\| x \|_2 = 1$ we have
Equation:

$$\min_{q \in Q_T} \| x - q \|_2 \leq \delta/14.$$

From [link], we can choose such a set Q_T with $\left|Q_T\right| \leq (42/\delta)^K$. We then repeat this process for each possible index set T, and collect all the sets Q_T together:
Equation:

$$Q = \bigcup_{T:|T|=K} Q_T.$$

There are $\binom{N}{K}$ possible index sets T. We can bound this number by
Equation:

$$\binom{N}{K} = \frac{N(N-1)(N-2)\cdots(N-K+1)}{K!} \leq \frac{N^K}{K!} \leq \left(\frac{eN}{K}\right)^K,$$

where the last inequality follows since from Sterling's approximation we have $K! \geq (K/e)^K$. Hence $|Q| \leq (42eN/\delta K)^K$. Since the entries of Φ are drawn according to a strictly sub-Gaussian distribution, from [link] we have [link]. We next use the union bound to apply [link] to this set of points with $\epsilon = \delta/\sqrt{2}$, with the result that, with probability exceeding
Equation:

$$1 - 2(42eN/\delta K)^K \, e^{-M\delta^2/2\kappa^*},$$

we have
Equation:

$$\left(1 - \delta/\sqrt{2}\right) \| q \|_2^2 \leq \| \Phi q \|_2^2 \leq \left(1 + \delta/\sqrt{2}\right) \| q \|_2^2, \quad \text{for all } \; q \in Q.$$

We observe that if M satisfies [link] then

Equation:

$$\log\left(\frac{42eN}{\delta K}\right)^{K} \leq K\left(\log\left(\frac{N}{K}\right)+\log\left(\frac{42e}{\delta}\right)\right) \leq \frac{M}{\kappa_1}+M\log\left(\frac{42e}{\delta}\right)$$

and thus [link] exceeds $1-2e^{-\kappa_2 M}$ as desired.

We now define A as the smallest number such that
Equation:

$$\| \Phi x \|_2 \leq \sqrt{1+A}, \quad \text{for all } \; x \in \Sigma_K, \; \| x \|_2 = 1.$$

Our goal is to show that $A \leq \delta$. For this, we recall that for any $x \in \Sigma_K$ with $\| x \|_2 = 1$, we can pick a $q \in Q$ such that $\| x-q \|_2 \leq \delta/14$ and such that $x-q \in \Sigma_K$ (since if $x \in X_T$, we can pick $q \in Q_T \subset X_T$ satisfying $\| x-q \|_2 \leq \delta/14$). In this case we have
Equation:

$$\| \Phi x \|_2 \leq \| \Phi q \|_2 + \| \Phi (x-q) \|_2 \leq \sqrt{1+\delta/\sqrt{2}} + \sqrt{1+A}\cdot \delta/14.$$

Since by definition A is the smallest number for which [link] holds, we obtain $\sqrt{1+A} \leq \sqrt{1+\delta/\sqrt{2}} + \sqrt{1+A}\cdot\delta/14$. Therefore
Equation:

$$\sqrt{1+A} \leq \frac{\sqrt{1+\delta/\sqrt{2}}}{1-\delta/14} \leq \sqrt{1+\delta},$$

as desired. We have proved the upper inequality in [link]. The lower inequality follows from this since
Equation:

$$\| \Phi x \|_2 \geq \| \Phi q \|_2 - \| \Phi (x-q) \|_2 \geq \sqrt{1-\delta/\sqrt{2}} - \sqrt{1+\delta}\cdot\delta/14 \geq \sqrt{1-\delta},$$

which completes the proof.

Above we prove above that the RIP holds with high probability when the matrix Φ is drawn according to a strictly sub-Gaussian distribution. However, we are often interested in signals that are sparse or compressible in some orthonormal basis $\Psi \neq I$, in which case we would like the matrix $\Phi\Psi$ to satisfy the RIP. In this setting it is easy to see that by choosing our net of points in the K-dimensional subspaces spanned by sets of K columns of Ψ, [link] will establish the RIP for $\Phi\Psi$ for Φ again drawn from a sub-Gaussian distribution. This *universality* of Φ with respect to the sparsity-inducing basis is an attractive property that was initially observed for the Gaussian distribution (based on symmetry arguments), but we can now see is a property of more general sub-Gaussian distributions. Indeed, it follows that with high probability such a Φ will simultaneously satisfy the RIP with respect to an exponential number of fixed bases.

ℓ_1 minimization proof

In this module we prove one of the core lemmas that is used throughout this course to establish results regarding ℓ_1 minimization.

We now establish one of the core lemmas that we will use throughout this course. Specifically, [link] is used in establishing the relationship between the RIP and the NSP as well as establishing results on minimization in the context of sparse recovery in both the noise-free and noisy settings. In order to establish [link], we establish the following preliminary lemmas.

Suppose are orthogonal vectors. Then

Equation:

We begin by defining the vector . By applying standard bounds on norms (Lemma 1 from "The RIP and the NSP") with , we have . From this we obtain

Equation:

Since and are orthogonal, , which yields the desired result.

If satisfies the restricted isometry property (RIP) of order , then for any pair of vectors with disjoint support,

Equation:

Suppose with disjoint support and that Then, and . Using the RIP we have

Equation:

Finally, applying the parallelogram identity
Equation:

establishes the lemma.

Let be an arbitrary subset of such that . For any vector , define as the index set corresponding to the largest entries of (in absolute value), as the index set corresponding to the next largest entries, and so on. Then
Equation:

We begin by observing that for ,
Equation:

since the sort to have decreasing magnitude. Applying standard bounds on norms (Lemma 1 from "The RIP and the NSP") we have
Equation:

proving the lemma.

We are now in a position to prove our main result. The key ideas in this proof follow from [link].

Suppose that satisfies the RIP of order . Let be an arbitrary subset of such that , and let be given. Define as the index set corresponding to the entries of with largest magnitude, and set . Then
Equation:

where
Equation:

Since , the lower bound on the RIP immediately yields
Equation:

Define as in [link], then since , we can rewrite [link] as
Equation:

In order to bound the second term of [link], we use [link], which implies that
Equation:

for any . Furthermore, [link] yields .
Substituting into [link] we obtain
Equation:

From [link], this reduces to
Equation:

Combining [link] with [link] we obtain
Equation:

which yields the desired result upon rearranging.